UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

TESIS DOCTORAL

ARQUITECTURAS Y MÉTODOS EN SISTEMAS DE RECONOCIMIENTO AUTOMÁTICO DE HABLA DE GRAN VOCABULARIO

JAVIER MACÍAS GUARASA
Ingeniero de Telecomunicación

2001

UNIVERSIDAD POLITÉCNICA DE MADRID
DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

TESIS DOCTORAL

ARQUITECTURAS Y MÉTODOS EN SISTEMAS DE RECONOCIMIENTO AUTOMÁTICO DE HABLA DE GRAN VOCABULARIO

JAVIER MACÍAS GUARASA
Ingeniero de Telecomunicación

Directores de la tesis
Dr. Ingeniero JOSÉ MANUEL PARDO MUÑOZ
Dr. Ingeniero JAVIER FERREIROS LÓPEZ

2001

Tesis Doctoral:	ARQUITECTURAS Y MÉTODOS EN SISTEMAS DE RECONOCIMIENTO AUTOMÁTICO DE HABLA DE GRAN VOCABULARIO
Autor:	JAVIER MACÍAS GUARASA
Directores:	Dr. INGENIERO JOSÉ MANUEL PARDO MUÑOZ Dr. INGENIERO JAVIER FERREIROS LÓPEZ

El tribunal nombrado para juzgar la tesis Doctoral arriba citada, compuesto por los doctores:

PRESIDENTE:	D. Ricardo de Córdoba Herralde
VOCALES:	D. Antonio José Rubio Ayuso
	D. Antonio Bonafonte Cávez
	D. Daniel Tapias Merino
SECRETARIO:	D. José Colás Pasamontes

acuerda otorgarle la calificación de SOBRESALIENTE CUM LAUDE (5 votos)

Madrid, 30 de Noviembre de 2001

El Secretario del Tribunal

A mi familia y a Sira

Agradecimientos

Este trabajo no habría sido posible sin el apoyo de distintas instituciones, grupos de investigación y muchas personas, tanto en el terreno profesional como el personal.

En primer lugar, a José Manuel Pardo y Javier Ferreiros, directores de este trabajo, por haberme dado aquella oportunidad en el año 90 (¿recuerdas tu llamada Javi?...), y por la enorme paciencia y el apoyo que me han dado a lo largo de estos años.

El Grupo de Tecnología del Habla me ha proporcionado todo su apoyo profesional, además de conseguir convertir en agradables los largos días de trabajo y ser capaces de seguir generando ilusión por nuestro tema. Gracias también a otros Centros o Departamentos con los que he tenido relación: esta tesis no habría sido la misma sin Telefónica I+D, sus proyectos, sus profesionales y la valiosa información que ha puesto a mi disposición y que espero haya dado sus frutos. También a la gente de DISAM por su capacidad de combinar esfuerzos y conseguir hacer funcionar las cosas; y al personal del CEAPAT, con los que compartimos la ilusión de trabajar en el apasionante campo de las tecnologías para discapacitados.

Gracias por supuesto a todos mis amigos y compañeros del Departamento de Ingeniería Electrónica de la ETSI de Telecomunicación de la UPM, que han compartido conmigo estos años y me han dado la oportunidad de disfrutar de un entorno académico e investigador de primera línea, y de los que tantísimo he aprendido.

Mi contacto con alumnos (ahora muchos ya son ex-) ha sido intenso y gratificante y de ellos he aprendido infinito. Gracias a we@thehackers; a los miembros del Grupo J&J y ADL y, más recientemente, al Wian Team. Gracias también a tantos instructores que han pasado por los labos, por hacerme la vida más fácil. También merecen ser mencionados los sucesivos “comejecs” de la Rama de la Escuela, con los que he trabajado desde el año 1997, y los miembros activos.

Gracias también a la familia del IEEE, que me ha permitido conocer otros mundos y aprender de ellos: a Levent, que me abrió las puertas; a Gozde, por su calidad humana; a Jens & Regina (ahora os toca a vosotros, chicos), a Martin, Kurt, Jordan, Christian y a sac-all (demasiados para nombrarlos aquí, disculpadme), en especial a Ed Palacios y Laura Durrett, así como a Cecelia Jankowski y, como no, Ted Hissey.

En el plano más personal, he tenido mucha buena gente cerca. Gracias llenas de cariño a mi gente de Málaga, a José de la Torre que ha sabido mantener el contacto; a Javi y Eva por los buenos y malos (algunos malísimos, ¿verdad?) ratos pasados y por seguir ahí; al resto de la panda, en especial a Inma y Estrella.

Quiero dar también las gracias a los que han compartido conmigo un techo y más de un plato: a Quino, Zalo & Co., Miguel Ángel, Jorge Sabater y Ana, y muy muy especiales a Ángel, Nuria y Laura, que me han alegrado estos últimos años. A Fernando González, por su saber hacer, su saber estar y por conservar los pies en el suelo y ayudarme a ver que no estamos solos. A Pepe, por su entusiasmo y por saber cómo transmitirlo y crear equipo. Gracias también a tantos esos del enemigo, por soportar lo suyo: Jorge, Jero, Javi Rubio (¡para que luego digas!), Jose y, sobre todo, Juan Ignacio.

No me puedo olvidar ni por asomo de mencionar con mayúsculas, negrita, subrayado y font 650 a Javi y Mónica, para los que las palabras que yo pueda decir aquí no serán nunca suficientes, ¡chapeau Amigos! porque lo habéis demostrado todo.

Por último, muchas gracias, gigantes, infinitas, a mi familia y a Sira, por su cariño y apoyo incondicional, por entenderme como soy (os habéis ganado el cielo) y por soportar todo el tiempo que no les he dado (tranqui Inés, ya estoy aquí). ¡Os quiero!

Por supuesto, son todos los que están, pero no están todos los que son. La primera versión de esta página decía: “Difícil tarea la del que tiene que agradecer todo a tantos sin olvidarse de ninguno...”. Con la seguridad de que no lo he conseguido ni de lejos, pido disculpas por las omisiones, fruto de mi olvidadiza cabeza.

Muchísimas gracias a todos

Resumen

La tesis que se presenta en este documento, se enmarca en el área del Reconocimiento Automático de Habla y específicamente en el diseño de sistemas de reconocimiento de gran vocabulario. En todos los casos, la tecnología de base en lo que se refiere al modelado, la aportan los modelos ocultos de Markov que, hoy por hoy, representan el paradigma de modelado dominante. En concreto, se utilizarán técnicas de modelado discreto y semicontinuo, dependiente e independiente del contexto.

En primer lugar, y a partir de una clasificación de alternativas arquitecturales en el diseño de sistemas de reconocimiento se hace un estudio teórico de la formulación del comportamiento de arquitecturas multi-módulo, tanto en coste computacional como en tasa de reconocimiento, definiendo una metodología de diseño para determinar la adecuación de módulos particulares de cara a su uso conjunto, que es validada con la experimentación correspondiente.

Igualmente, se hace énfasis en el estudio y evaluación de algunas de las alternativas de compresión del espacio de búsqueda, estableciendo relaciones de compromiso entre coste y tasa, que es el binomio decisivo a la hora de abordar el diseño de sistemas en tiempo real. Se presentan estudios sobre distintas estrategias de organización del espacio de búsqueda orientadas a exploración y búsqueda con algoritmos de programación dinámica: árboles y grafos, deterministas y no deterministas, proponiendo soluciones prometedoras para incrementar la tasa de inclusión obtenible sobre estructuras de grafo (en las que la compresión del espacio de búsqueda produce peores resultados que con la búsqueda lineal o en árbol). Especialmente importante es el trabajo sobre estimación de listas variables de preselección, analizando métodos paramétricos y no paramétricos, centrándonos en el uso de redes neuronales como mecanismo estimador. Se ha propuesto una metodología de selección de parámetros de entrada, topologías y métodos de codificación, en base a su potencia discriminativa en una tarea simplificada. Dicha propuesta que ha sido ampliamente evaluada y comparada con el enfoque tradicional de uso de listas fijas, mostrando la consistente mejora tanto en tasa como en coste computacional conseguible con el uso de redes neuronales. Dicho estudio sobre listas variables ha sido extendido de forma natural al problema de estimación de fiabilidad de hipótesis, habiéndose aprovechando estos resultados, de nuevo, para la estimación de longitudes de listas, obteniendo también buenos resultados.

En lo que respecta al repertorio de unidades de reconocimiento y a la composición de los diccionarios usados (en cuanto al uso de múltiples pronunciaciones), se aplican, evalúan y comparan métodos dirigidos por datos y basados en conocimiento. En el apartado de introducción de variantes de pronunciación se ha discutido ampliamente la problemática de contar con bases de datos representativas y haciendo énfasis en la importancia de atender y evaluar las mejoras marginales obtenidas con algunos de estos métodos.

La evaluación de los resultados es planteada cuidadosamente, sobre dos tareas radicalmente distintas: habla telefónica independiente del locutor y habla aislada dependiente, ambas usando gran vocabulario (hasta 10000 palabras), lo que permite obtener conclusiones y claves de diseño para cada una de ellas, con lo que se consigue una generalización más fundamentada de su bondades o perjuicios. En este sentido se aplican análisis de validez y relevancia estadística que pongan en su justo sitio las mejoras o degradaciones observadas. En los procesos de evaluación se han propuesto nuevas métricas y mecanismos originales de comparación.

Abstract

This Ph.D. thesis is centered in the automatic speech recognition area and, specifically, in the design of large vocabulary speech recognition systems. In our case, the acoustic modeling technology is based on Hidden Markov Models, both in discrete and semicontinuous versions with both context dependent and independent units.

In first place, and starting from a classification of architectural alternatives in the design of speech recognition systems, we made a theoretical study of the behavior of multi-module architectures, both in computational cost and recognition rate, defining a design methodology that allows us to determine the adequacy of any given speech recognition modules when being used in a joint manner. This study is validated with the corresponding experiments.

We also make special emphasis in the study and evaluation of different alternatives in order to compress the search space, establishing trade-off relations between computational cost and recognition rate, being these factors the ones to be taken into account when designing real-time systems. We show studies on different strategies to organize the search space oriented to guide the search with dynamic programming algorithms: trees and graphs, deterministic or not, proposing promising solutions to increase the inclusion rate achievable when using graph structures (in which the search space compression leads to worse results than the ones obtained with linear or tree-based spaces). Specially important is the work done on variable preselection list estimation, analyzing parametric and non-parametric methods, and further studying the use of neural networks as the estimation mechanism. We have proposed a methodology to select the input parameters along with the topologies and coding methods to be used, according to their discrimination capabilities in a simplified task. This proposal has been widely evaluated and compared with the traditional approach of using fixed length lists, showing a consistent improvement, both in rate and computational cost, when using the neural network based strategy. This study on variable length preselection lists has been extended, in a natural way, to the problem of confidence estimation. The estimated confidence values are then used again in the preselection list length estimation, getting also good results.

In which respect to the inventory of recognition units and the composition of the dictionaries used (using multiple pronunciations), we apply, evaluate and compare different methods, both data-driven and knowledge-based. In the chapter addressing the introduction of pronunciation variants, we have deeply discussed the need to have representative databases, making special emphasis in the importance of evaluating the marginal improvements achievable with some of these methods.

The evaluation is carefully planned, on two radically different large vocabulary (10000 words) isolated word recognition tasks: speaker independent telephone speech and speaker dependent clean speech, which allows us to obtain conclusions and design guidelines for each of them, leading to a better generalization of the advantages and disadvantages of every method. In this sense we apply statistical significance tests to put in the right place the observed improvements or degradations. In the evaluation process, new metrics and original comparison methods have been proposed.

Glosario

Se incluyen en este glosario algunas breves definiciones, útiles para la lectura del documento:

Tasa de inclusión/preselección:

Tasa de reconocimiento obtenida en la etapa de hipótesis considerando un cierto número de candidatos (*l*). Nos referiremos a ella como $tasaInclusion(l)$. La representación de la tasa de inclusión en función del número de candidatos usados nos dará la curva de tasa de inclusión para la tarea considerada.

Tasa de error de inclusión/preselección:

Tasa de error obtenida en la etapa de hipótesis, considerando un cierto número de candidatos (*l*). Nos referiremos a ella como $tasaErrorInclusion(l)$ Obviamente:.

$$tasaErrorInclusion(l) = 100 - tasaInclusion(l)$$

Longitud de lista (o esfuerzo):

Longitud de la lista de preselección para una palabra a reconocer dada (*i*). Nos referiremos a ella como $longLista(i)$. Dicha lista será la que el sistema de preselección entregue al módulo de análisis detallado. La denominación de *esfuerzo* procede de la relación que tiene la longitud de lista con la demanda computacional de la etapa de análisis detallado, refiriéndonos fundamentalmente a tiempo de proceso.

Esfuerzo total:

Definido como la suma de todos los esfuerzos particulares, es decir, la suma de las longitudes de lista preselección usadas para cada palabra en una tarea dada. Nos referiremos a él como $EsfuerzoTotal$:

$$EsfuerzoTotal = \sum_{i=1}^{NumPalBD} longLista(i)$$

Donde *NumPalBD* es el número de palabras a procesar de las que consta nuestra base de datos.

Esfuerzo medio:

Definido para una tarea de reconocimiento determinada (básicamente para un conjunto de palabras a procesar), es la longitud media de la lista de preselección para todas las palabras. Nos referiremos a él como $esfuerzoMedio$, y vendrá dado por:

$$esfuerzoMedio = \frac{EsfuerzoTotal}{NumPalBD}$$

Posición de acierto:

Definida como la posición en la lista de preselección en la que se reconoció una palabra dada *i* dentro de la lista de preselección. Nos referiremos a ella como $posicOK(i)$

Índice de Contenidos

Índice de Figuras

Índice de Figuras (cont.)

Índice de Figuras (cont.)

Índice de Figuras (cont.)

Índice de Figuras (cont.)

Índice de Figuras (cont.)

Índice de Figuras (cont.)

Índice de Figuras (cont.)

Índice de Tablas

Índice de Tablas (cont.)

Índice de Tablas (cont.)

Índice de Tablas (cont.)

1 Introducción

1.1 Presentación

La tesis que se presenta en este documento, se enmarca en el área del Reconocimiento Automático de Habla (RAH), que, básicamente, plantea como objetivo fundamental la transcripción automática de la señal de voz mediante máquinas, la conversión de habla en texto.

El habla es sin duda el método de comunicación más intuitivo y natural para los seres humanos. La Tecnología del Habla, en sentido general, está gozando de un interés cada vez más creciente por parte, no sólo de la comunidad científica internacional, sino de la sociedad en general y, entre ellas, el reconocimiento automático de habla presenta uno de los campos más atractivos de investigación.

Gracias a los tremendos avances en este campo, la imagen del ordenador *HAL9000* en la película *2001: Una odisea del espacio*, dirigiéndose a la tripulación con una voz de alta naturalidad e inteligibilidad y, sobre todo, atendiendo a las órdenes de aquélla, con conversaciones absolutamente espontáneas, se ve cada vez más cerca.

Hasta hace pocos años, los sistemas de reconocimiento automático de habla (SRAH) de mediana y gran complejidad (en cuanto a las tareas que abordaban), estaban disponibles casi únicamente como prototipos de laboratorio. En la actualidad, empresas como IBM, Philips, Lernout & Hauspie y Dragon Systems han abordado el terreno del mercado de gran consumo con productos de elevada calidad para dictado de textos, tanto en habla aislada como continua. Por otro lado, empresas como, por ejemplo, Telefónica I+D, Philips, Lernout & Hauspie, AT&T, Unisys, NUANCE, SpeechWorks, incluso IBM y Dragon Systems, ofrecen sistemas de suministro de información por línea telefónica de creciente habilidad para llevar a buen término un alto porcentaje de las solicitudes y transacciones requeridas por los usuarios; y cada día son más las empresas que abren (o crean) sus departamentos de I+D para explotar esta tecnología.

La intención fundamental de esta tesis es profundizar en varios puntos de importancia sobre los que no hay conclusiones claras y definitivas, a la hora de diseñar sistemas de reconocimiento de gran vocabulario.

De entrada, podemos pensar en varias aproximaciones al diseño inicial del sistema, a su arquitectura. Si nos centramos en el módulo de búsqueda, en nuestro caso, plantearemos dos líneas básicas: sistemas integrados, en los que el proceso de búsqueda utiliza simultáneamente todas las fuentes de información disponibles y suele hacerse en un único paso, con mecanismos de guiado; y sistemas no integrados, en los que la tarea se divide en sub-procesos de menor complejidad, accediendo cada uno de ellos a algunas de las fuentes de información disponibles. Los primeros presentan un mayor coste computacional, con la ventaja de producir, generalmente, mejores resultados; mientras que los segundos presentan el efecto contrario. En cada caso, se tratará de buscar el equilibrio entre ambos factores, como se discutirá más adelante. Igualmente se abordará el estudio de criterios de diseño en reconocedores multi-módulo.

En lo que respecta a detalles más concretos de implementación, en la literatura pueden encontrarse multitud de aproximaciones y enfoques para abordarlos. Un problema fundamental a la hora de estimar la bondad de una técnica determinada es que se suele presentar centrada en una tarea o aplicación concreta. Así, es difícil extrapolar hasta qué punto dicha técnica puede suponer mejoras sustanciales en otras de distinta configuración. En esta tesis, se pretenden abordar distintos sistemas y distintas tareas, sobre los que la experimentación de una serie de metodologías permitirá obtener conclusiones y claves de diseño para cada uno de ellos, con lo que se conseguirá una generalización más fundamentada de las bondades o perjuicios de cada uno.

También en ocasiones, algunas de las decisiones de diseño de los sistemas de reconocimiento automático de habla (SRAH) están basadas en criterios proporcionados por expertos humanos,

fundamentadas en conocimiento lingüístico o en experiencia empírica previa. Las fuentes de información generalmente utilizadas suelen extraerse directamente de los datos de entrenamiento, pero detalles más concretos como, por ejemplo, el número y tipo de los modelos acústicos utilizados, la composición o derivación de los modelos léxicos, la topología usada en un modelo de Markov, o en una red neuronal, o el conjunto de parámetros generados por el preprocesador, son todos ellos muestras de decisiones y criterios definidos por el investigador o el ingeniero, de nuevo, basándose en un alto grado de conocimiento sobre la tarea, pero tratándose por tanto de fuentes de conocimiento estático (en el sentido de que son prefijados y no se modifican a partir de los datos de entrenamiento), en contraposición a, por ejemplo, los parámetros de los modelos acústicos generados.

En nuestro caso, será de interés lo que respecta a la selección de las unidades de reconocimiento y la selección de los esquemas alofónicos de las palabras del diccionario: en el idioma castellano, un sistema de conversión grafema-fonema proporciona una calidad razonable, pero es inviable en otros idiomas, debiéndose recurrir a extensos diccionarios de elaboración manual o, como mucho, semiautomática. En cualquier caso, incluso en castellano, no es razonable asumir ninguna pronunciación *estándar* como la única válida, si estamos pensando en extender el sistema para un conjunto amplio de usuarios (y, por tanto, de condiciones desconocidas en cuanto a acento, formación cultural, etc.).

Así, en esta tesis plantearemos soluciones que buscan la elaboración automática tanto del repertorio de unidades sub-léxicas, como de las entradas en los diccionarios de pronunciaciones de cada aplicación. En cada caso se buscarán soluciones *dirigidas por datos* (*data-driven*), es decir, las propiedades del sistema se aprenderán a partir de un conjunto de datos de entrenamiento; y también las *basadas en conocimiento* (*knowledge-based*), proponiendo la combinación de ambas como la solución más eficaz.

Profundizando en esta línea, y como tema lateral, se pretende estudiar con detalle el efecto real del incremento de tamaño del diccionario sobre el rendimiento de un sistema. Por una parte, en lo que a coste computacional se refiere (punto éste que enlaza con el que se comentará a continuación); y por otra en lo que atañe a tasa de reconocimiento, tratando de independizar al máximo dicha medida de las características intrínsecas del diccionario en sí, lo que llevará a conclusiones de interés sobre la *independencia del vocabulario*.

Un aspecto que no puede obviarse es la complejidad algorítmica que pone en juego un SRAH, ya que influye directamente en la posibilidad de llegar a su implementación en tiempo real, objetivo último desde un punto de vista de ingeniería. El aligerar la demanda computacional de cualquier proceso, permitirá avanzar en la aplicación práctica de técnicas cada vez más complicadas (y costosas) que aborden tareas de crecientes órdenes de dificultad. Así, en el desarrollo de esta tesis se hará énfasis en el análisis de algunas de las alternativas de optimización algorítmica, de reducción de complejidad y de reducción del espacio de búsqueda, estableciendo relaciones de compromiso entre coste y tasa, que es el binomio decisivo a la hora de abordar diseños de sistemas en tiempo real[1].

Resumiendo, las ideas fundamentales en las que se centrará esta tesis son las siguientes:

- Alternativas y rendimiento de arquitecturas integradas y no integradas del reconocedor. Estudio de criterios de diseño en reconocedores multi-módulo
- Optimización algorítmica y reducción de complejidad
- Selección de las unidades de reconocimiento y selección del diccionario: múltiples pronunciaciones.

1. Téngase en cuenta, de entrada, que el factor fundamental es la tasa de reconocimiento obtenida, de modo que la obtención de un sistema de calidad prima absolutamente sobre la obtención de un sistema veloz. Siempre se podrán plantear optimizaciones sobre un sistema caro computacionalmente, pero un sistema barato de entrada no garantizará nunca un rendimiento adecuado, o, como poco, la posibilidad de crecer en calidad (al menos fácilmente).

En todos los casos, la tecnología de base en lo que se refiere al modelado, la aportan los modelos ocultos de Markov que, hoy por hoy, representan el paradigma de modelado dominante. En concreto, se utilizarán técnicas de modelado discreto y semicontinuo, dependiente e independiente del contexto.

Como comentamos más arriba, para acotar lo más posible la validez de los resultados y conclusiones que se obtengan, se planifica la aplicación de las técnicas y metodologías estudiadas y propuestas sobre dos tareas diferentes, ambas de habla aislada, y ambas para grandes vocabularios (hasta 10000 palabras):

1. Independencia del locutor: En una tarea de reconocimiento de nombres propios por línea telefónica.
2. Dependencia del locutor: Diseñada originalmente para su uso en una tarea de dictado de textos sin restricción.

El interés de ambas radica en que corresponden a dos de las áreas más prometedoras de aplicación de los SRAH: sistemas de información por línea telefónica y sistemas de dictado. Adicionalmente, sus particularidades en cuando a demanda de modelado y restricciones en la población de usuarios, los hacen lo suficientemente distintos como para poder aportar interesantes conclusiones sobre las técnicas que se aplicarán.

Dada la sustancial diferencia entre las tareas base de reconocimiento que se abordarán en la tesis, y las alternativas planteadas en el estudio, una idea central será establecer los límites de validez y aplicabilidad de cada uno de los métodos desarrollados, para lo que se desarrollará un cuidadoso plan de experimentos (proponiendo las métricas y estrategias de comparación que se consideran más adecuadas), que se irá perfilando a medida que avance el trabajo, aplicando en todos los casos pruebas y análisis de validez y relevancia estadística que pongan en su justo sitio las mejoras o degradaciones observadas.

1.2 Estructura del documento

El presente documento está estructurado en 7 capítulos. El primero es esta introducción y en el segundo se describe el encuadre científico-tecnológico que enmarca nuestro trabajo. En el tercero se aborda el estudio de alternativas arquitecturales para pasar en el cuarto a discutir aspectos relacionados con la compresión del espacio de búsqueda y la reducción de carga computacional. En el quinto se trata el problema de la selección de unidades de reconocimiento y de entradas en los diccionarios con la introducción de múltiples pronunciaciones. El sexto y séptimo presentan las conclusiones y líneas futuras más destacadas de este trabajo, respectivamente, y finalmente se incluyen una serie de apéndices con temas relacionados de interés.

Dada la diversa temática abordada en esta tesis, se ha optado por incluir en cada capítulo un apartado destinado a recoger, de forma más detallada que en el sexto, las conclusiones particulares aplicables.

2 Encuadre científico-tecnológico

A pesar de los avances comentados en el Capítulo 1, el objetivo último de los sistemas de reconocimiento automático de habla, i.e., la transcripción sin errores de habla espontánea está aún lejos de ser alcanzada. Por supuesto, no podemos dudar de que las técnicas de RAH están en un momento de razonable madurez, y los sistemas en el mercado prueban que, efectivamente, es útil, pero hay un cierto número de preguntas que aún no tienen respuestas definitivas. Una de las intenciones de esta tesis es profundizar en algunas de ellas.

En este capítulo daremos referencias al estado actual de la cuestión en algunos de los temas que pretendemos abordar en la tesis, para situar éstos en su contexto adecuado.

2.1 Modelado

Como ya hemos señalado, utilizaremos como paradigma de modelado acústico los modelos ocultos de Markov (HMM), que constituyen la técnica más utilizada en los laboratorios de todo el mundo, desde los trabajos pioneros de Baker [Baker75] y Jelinek [Jelinek76].

Un HMM resulta de la composición de dos procesos estocásticos, en las que la secuencia de unidades de reconocimiento subyacente y las observaciones acústicas están modeladas como procesos de Markov. Dicho de otro modo, se modela, por una parte, la secuencia temporal y, por otra, los eventos acústicos que se producen en dicha secuencia. No entraremos en detalle de la formulación de los HMM, remitiendo al lector a la introducción presentada en [Rabiner86], o al tutorial en [Rabiner89a]. Para una descripción más pormenorizada, podemos referenciar a [Huang90b].

La evidencia resultante de su uso intensivo durante más de 20 años, muestra que los HMM son lo suficientemente potentes como para modelar adecuadamente la mayor parte de las fuentes de variabilidad presentes en el habla, a pesar de la existencia de ciertos problemas bien conocidos: discriminación relativamente pobre, requerimientos explícitos en las asunciones sobre las distribuciones utilizadas, la no consideración de la correlación entre vectores acústicos, falta de adecuación en el modelado temporal, etc. [Lee89a] [Morgan95].

Dentro de la formulación genérica de los HMM, podemos hacer una clasificación en función de la naturaleza de las distribuciones que modelan las observaciones acústicas. Así, en una primera aproximación, podemos definir dichas distribuciones en un espacio discreto de símbolos, dando lugar a los *Modelos discretos de Markov* (DDHMM) [Huang90b] [Hasan90]. En este caso, las observaciones acústicas son símbolos pertenecientes a un alfabeto finito, con lo que se utilizan técnicas de cuantificación vectorial para transformar el vector de parámetros de entrada en uno de esos símbolos finitos [Gray84] [Rabiner86].

Análogamente, podemos definir las distribuciones de probabilidad de las observaciones en un espacio continuo, dando lugar a los *Modelos Continuos de Markov* (CDHMM) [Soudoplatoff86] [Huang90b]. En este caso, es necesario aplicar ciertas restricciones para limitar la complejidad de los procesos de estimación y cálculo de las probabilidades asociadas. El mecanismo más usual caracteriza cada modelo como una mezcla de funciones del mismo tipo (generalmente gausianas). Esta aproximación puede tener problemas de estimación fiable de sus parámetros y de demanda computacional, aunque se han desarrollado diversas técnicas para aliviarlos, basadas fundamentalmente en la reducción del número de parámetros a estimar (generalmente compartiendo distribuciones o combinando modelos), en base a distintos criterios [Lee90] [Fissore91] [Hwang93a] [Hwang93b], y en estrategias de optimización de cálculo, comentadas en el Apartado 2.3.

Por último, y en la misma línea de reducir los inconvenientes que presentan tanto los DDHMM como los CDHMM, surgen los Modelos Semicontinuos de Markov (SCDHMM) [Huang89] [Huang90], en los que se comparte el mismo conjunto de funciones de densidad de probabilidad para distintos modelos, variando únicamente los pesos de ponderación aplicados a cada una de ellas, lo que puede verse como una unificación de los dos enfoques previos. Comparándolo con la versión discreta,

este modelado hace más exacto el proceso de cuantificación y robustece la estimación de probabilidad, al considerar varias distribuciones en cada caso, permitiendo además la estimación conjunta de los parámetros de la cuantificación y del modelo en sí. Comparándolo con la continua, reduce el número de distribuciones a considerar, robusteciendo la estimación en el entrenamiento, aunque conservando la capacidad de modelado a partir de la mezcla de aquellas [Huang93].

Dado que los HMM no modelan con precisión la duración de las unidades de que se trate [Rabiner89b], es posible incluir información sobre ella en el algoritmo. A lo largo del tiempo se han descrito en la literatura métodos para incluir explícitamente funciones de densidad de duración de estado en los HMM ([Rabiner89b], [Cifuentes91], [Kenny92], [Deng88] y [Levinson86]). Estudios previos en nuestro Grupo ([Macías92]) trataban las duraciones a nivel de modelo completo. Dicha información sobre las duraciones de las unidades se extraía durante los procesos de entrenamiento de los HMM, y se asumió que la distribución de la duración atribuida a una unidad es gausiana. Los resultados obtenidos no fueron significativamente mejores, con lo que se abandonó esa línea para los trabajos de esta tesis.

La segunda alternativa en éxito y utilización de cara a abordar el modelado acústico en RAH, la constituyen las redes neuronales (NN) [Morgan91], que, básicamente, son estructuras con capacidad de clasificación y discriminación, intrínsecamente no lineales, pudiendo aprender una determinada tarea a partir de pares observación-objetivo, sin necesidad de hacer ningún tipo de suposición sobre el modelo subyacente. Son esas dos características las que las hacen tan atractivas para su aplicación en SRAH. Los inconvenientes fundamentales son el desconocimiento a priori de la topología adecuada a una cierta tarea; el elevado tiempo de entrenamiento que requieren; su dificultad para el modelado temporal y la posibilidad de quedar estancadas en puntos inadecuados del entrenamiento (mínimos locales de las funciones de error). En cualquier caso, se han desarrollado arquitecturas y soluciones concretas para reducir en lo posible sus defectos en el modelado temporal (redes neuronales recurrentes (RNN) [Robinson91] y de retardo temporal (TDNN) [Lang90]) y el coste de entrenamiento [Menéndez94a] [Menéndez94b]. En los últimos años, enfoques híbridos basados en HMM y NN han mostrado éxitos notables [Bourlard93] [Renals94] [Menéndez94a] [Hochberg94] [Robinson95] [Morgan95].

En este trabajo, utilizaremos en todos los casos HMM, en sus variantes discreta y semicontinua, basándonos en los sistemas y soluciones desarrolladas previamente en nuestro grupo [Macías94a] [Macías96] [Córdoba95] [Ferreiros96], ya que no es nuestro objetivo profundizar en técnicas de modelado, aunque se extraerán conclusiones sobre el uso de uno u otro tipo.

Por último, es importante hacer notar que para paliar los posibles problemas de falta de entrenamiento de los parámetros de los HMM es necesario aplicar técnicas de suavizado (*smoothing*). En la literatura se describen varias sobre el particular, como por ejemplo el suavizado umbral [Rabiner89a], el más simple y cuyo principal problema es su incapacidad de distinguir los casos improbables de los imposibles [Lee88]; el método de la co-ocurrencia ([Schwartz84), que usa información de qué símbolos son frecuentemente substituidos por otros (en otras palabras, cuando se observa un símbolo, cómo de probable es el hecho de que se observen los otros en contextos similares); el de la *distancia*, en el que durante el entrenamiento se generan múltiples símbolos de salida y se le asignan probabilidades basadas en sus distancias al vector de entrada y una ventana de Parzen ([Rabiner89b], [Hassan90]), etc. Otra técnica que ha dado excelentes resultados y en la que no vamos a entrar se trata en [Schwartz89] y [Bahl85] (*Deleted Interpolation*), planteándose estas posibilidades para desarrollos futuros.

2.2 Arquitecturas

Cuando nos enfrentamos a la toma de decisiones en el proceso de diseño de un SRAH, una de las primeras preguntas a responder es la que atañe a la especificación de la arquitectura modular del sistema.

En la Figura 2-1 se muestra el esquema genérico de un SRAH, pero dada la generalidad del mismo, de cara a la toma de decisiones, debemos profundizar un poco en detalles de diseño y tratar de hacer una clasificación que abarque en lo posible elementos importantes en la arquitectura global. Esta clasificación afecta fundamentalmente al módulo de búsqueda, que es en el que nos centraremos en este apartado.

Sin embargo, comenzaremos comentando brevemente algunas generalidades sobre el módulo de preproceso.

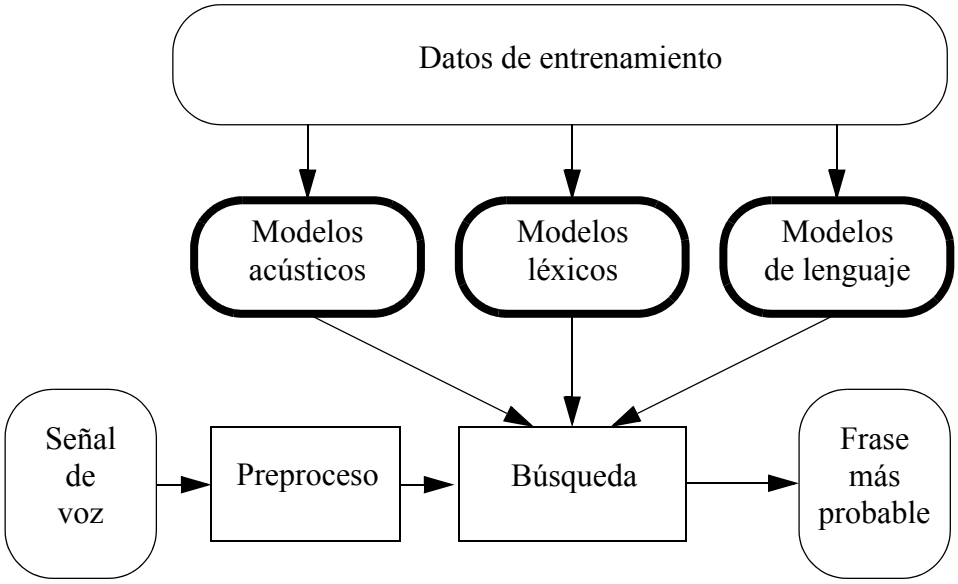

Figura 2-1: Esquema genérico de un SRAH

En RAH, la señal de entrada se representa generalmente por una secuencia temporal de vectores de parámetros, que son calculados mediante análisis localizado. En el caso general, la idea final es transformar dicha representación inicial en un conjunto final de parámetros, más elaborados, en el sentido de estar más adaptados a la tarea de discriminación posterior. La selección de qué parámetros sigue siendo un problema de difícil solución [Bourlard96], aunque la característica común a la mayoría de los enfoques actuales, pasa por un proceso de filtrado de dicha secuencia [Soong86]. En [Nadeu97] y [Chengalvarayan97] pueden encontrarse desarrollos orientados a la generalización de la formulación del proceso de parametrización. En nuestro caso, utilizaremos como punto de partida sistemas de preproceso relativamente clásicos, basados en cálculo de parámetros *mel*, sus derivadas, e incorporando versiones con filtrado RASTA [Hermansky94] [Córdoba95].

Ahora sí, atendiendo al módulo de búsqueda acústica y tratando de buscar una perspectiva arquitectural general, podremos distinguir entre:

- Arquitecturas basadas en el paradigma *hipótesis-verificación*, en las que, conceptualmente, hay dos o más pasos de reconocimiento, es decir, se descompone la tarea en varios procesos en cascada, en las que se opera sobre un conjunto cada vez más reducido de hipótesis, utilizando modelos cada vez más potentes. En general, cada una de esas sucesivas etapas se caracteriza por:

- Complejidad creciente: El coste computacional implicado para evaluar cada candidato es mayor a medida que avanzamos en la cadena de módulos de reconocimiento en cascada, debido a la mayor complejidad de los modelos y demás fuentes de información utilizadas.
- Resultados intermedios: Cada etapa ofrece a la siguiente un conjunto cada vez más reducido de alternativas entre las que decidir. Así, el incremento de complejidad del modelado se ve compensado por una menor demanda de cálculo al tener que operar sobre subconjuntos del espacio de alternativas inicial.

En estos sistemas cada etapa ofrece una hipótesis a la siguiente, que *verificará* dicha hipótesis, refinando el resultado, para proceder de la misma manera con la siguiente, o bien arrojará la decisión final, si es el último elemento de la cadena. Por supuesto debe asegurarse que la tasa obtenida por las etapas iniciales no *limita* significativamente la tasa final del sistema, y que la complejidad computacional final es menor que la que se obtendría con un sistema en un único paso. Así, habrá que jugar con los parámetros: tamaño de la lista de hipótesis vs. complejidad computacional de la etapa dada. En SRAH para gran y muy gran vocabulario, éste es el enfoque más usual [Cole95], siendo la estructura intermedia generada una malla de las *n* mejores hipótesis de la secuencia reconocida [Aubert94], o incluso un grafo de palabras [Ortmans97c], utilizando para ello modelos de poca resolución acústica.

- Arquitecturas en las que, conceptualmente, es necesario un único proceso en bloque, desde la señal de voz hasta la obtención de la transcripción de la misma a la salida, a las que podríamos llamar basadas en el paradigma de *verificación en un solo paso* o, simplemente, *verificación*. Este enfoque implica, genéricamente, un mayor coste computacional, al tener que operar sobre el conjunto total de alternativas del espacio de búsqueda, pero suelen ofrecer mejores resultados finales y deben ser, obviamente, más robustos que los anteriores. Por supuesto, es posible introducir mecanismos de limitación del mismo, para reducir su coste, pero la idea central es el proceso en un único paso, sin que aparezcan listas de candidatos intermedias.

La clasificación vista no impone diferencias significativas en cuando a la algorítmica implicada, en el sentido de que cada algoritmo podrá utilizarse en un módulo de hipótesis o de verificación en función de su mayor o menos robustez. En nuestro caso, y profundizando en una clasificación ortogonal a la anterior en la que nos basaremos en las distintas aproximaciones a la hora de utilizar las fuentes de conocimiento disponibles, podremos hablar de:

- *Sistemas no integrados*, en los que cada uno de los módulos implicados utiliza parte de la información disponible para realizar la búsqueda. Un ejemplo de ello es el representado por los sistemas que extraen una cadena o malla de unidades elementales, seguidas por un módulo de acceso léxico (de los que hay abundantes ejemplos en la literatura, como por ejemplo [Hood91], [Riley91], [Antoniol90], [Fissore89], [Ljolje92], [Gravier97], [Lee97] etc.) que decide las hipótesis finales que con mayor probabilidad corresponderían a la cadena/malla de entrada. El módulo de generación de cadena o malla solamente utiliza información del modelado acústico, y es el de acceso léxico el que finalmente impone las restricciones extraídas de un diccionario. Como ejemplos de sistemas basados en el paradigma hipótesis-verificación en este apartado, podemos nombrar los descritos en [Billi89], [Ney91], [Buttafava90] y [Macías94a, Macías96]. En el caso de sistemas basados en verificación, un ejemplo representativo es el descrito en [Mattsuura88].
- *Sistemas integrados*, en la que el resultado final (lista de preselección o candidato reconocido) se obtiene de forma directa, de modo que todas las fuentes de información disponibles se utilizan (se integran) simultáneamente en el proceso de búsqueda. Este enfoque implica, de nuevo en general, mayor coste computacional que el anterior, ya que una división en varios módulos suele suponer una reducción de aquél; pero ofrecen mejores resultados al posibilitar la introducción de

mecanismos de guiado de la búsqueda en el proceso desde el principio. Evidentemente, un sistema basado en hipótesis-verificación nunca podrá ser integrado ya que las fuentes de información se usan, forzosamente, de forma no simultánea. Ejemplos de este tipo de sistemas se describen en [Deng90] y [Aktas91].

Es esta última clasificación en la que centraremos nuestro estudio, al ser la que impone las mayores restricciones en lo que a algorítmica se refiere, ya que el enfoque integrado podrá utilizarse en un sistema basado en el paradigma de verificación siempre que las tasas de reconocimiento que consiga sean lo suficientemente altas para ser útiles en la tarea planteada. En caso contrario, siempre podrá utilizarse como módulo de generación de hipótesis en un sistema basado en hipótesis-verificación.

En lo que respecta a la algorítmica de la búsqueda acústica en sí, la aproximación más usual procede del campo del reconocimiento estadístico de patrones [Schalkoff92]. En general, y dada la característica secuencial del proceso del habla, muchos de los problemas en SRAH pueden resolverse aplicando técnicas de programación dinámica [Bellman57], en las que se persigue la determinación de un camino óptimo, o camino de mayor probabilidad (o menor coste), mediante la aplicación secuencial de optimizaciones (decisiones) locales, a lo largo de un espacio de búsqueda. El primer gran impulso algorítmico en SRAH vino con el desarrollo y aplicación de las técnicas de alineamiento dinámico temporal (*Dynamic Time Warping*, DTW) [Sakoe78], en la que el objetivo es hacer una comparación entre un patrón de voz y una producción desconocida, de forma no lineal, permitiendo solucionar el problema difícilmente tratable hasta ese momento de la diferencia en longitud entre los patrones y las producciones (por la variabilidad temporal inherente al proceso del habla). En la actualidad, es una técnica en desuso, por sus problemas de generalización, fundamentalmente. En su lugar, la misma base teórica de los algoritmos de programación dinámica, hace uso de la robustez del modelado estocástico (paramétrico) de los HMM para acometer tareas de todo tipo.

En nuestro caso, para la parte acústica, aplicaremos las ideas propuestas sobre una base previa de sistemas de reconocimiento de habla conectada disponibles en nuestro laboratorio, utilizando la idea de construcción/reconocimiento de modelos de secuencias más complejas (frases o palabras) a partir de la concatenación de modelos de unidades más simples (alófonos, por ejemplo) [Bridle79] [Bridle82] [Ney84].

Igualmente utilizaremos técnicas de acceso al léxico en los sistemas no integrados, básicamente fundamentados en los mismos principios de programación dinámica, pero aplicados a secuencias alfanuméricas, en lugar de vectores acústicos [Fissore89] [Macías94b].

2.3 Complejidad algorítmica

Los algoritmos empleados en sistemas de reconocimiento, como en cualquier otra tarea de reconocimiento estadístico de patrones, suelen ser exigentes en sus demandas computacionales, tanto en proceso de cálculo como en requisitos de memoria.

Cuanto más compleja sea la tarea a abordar, mayores serán dichos requerimientos, de modo que es fundamental plantear técnicas específicas que tiendan a reducirlos, sobre todo si el objetivo final es la implementación en tiempo real de sistemas de reconocimiento de gran vocabulario.

De acuerdo con la estructura general de cualquier sistema de reconocimiento, como la mostrada en la Figura 2-1, podemos pensar en optimizaciones específicas en cada uno de los módulos fundamentales: preproceso o búsqueda.

A grandes rasgos, las aproximaciones planteadas al respecto en la literatura, enfocan el problema desde tres puntos de vista:

- Optimización algorítmica, en la que se centran en la consecución de algoritmos más eficientes computacionalmente, bien mediante optimizaciones en los cálculos, independientemente de la implementación final, o bien mediante optimizaciones específicas de la implementación sobre la arquitectura hardware utilizada. En el primer caso, tenemos ejemplos muy claros en el terreno del procesado digital de la señal, en el

que hay numerosos trabajos al respecto. El segundo tiene mucho interés desde el punto de vista ingenieril, y responde en general a tareas propias de la etapa final del desarrollo de un sistema orientado a su uso en condiciones reales, en un producto comercial, por ejemplo.

- Reducción de complejidad, entendiendo con ello la búsqueda de qué elementos es posible eliminar de los cálculos, sin llegar a perjudicar la tasa de reconocimiento y [SanSegundo97] [López98]. Así por ejemplo, podemos citar trabajos de optimización en la búsqueda del centroide más próximo en un proceso de cuantificación vectorial [Bocchieri93], reducción de los parámetros con los que trabajar [Hunt89] [Haeb94], estimación aproximada de valores de gausianas, [Beyerlein94] [Córdoba95] [Watanabe95] [Knill96] [Ortmanns97a], etc.
- Reducción del espacio de búsqueda, que es donde mayor esfuerzo se ha invertido tradicionalmente, ya que en sistemas de gran vocabulario (objetivo último en esa dimensión), dicho espacio puede llegar a ser extremadamente grande. Las técnicas más usuales pasan por la limitación del espacio de búsqueda a una zona que previsiblemente contendrá la solución óptima [Ferreiros96], la implementación de dichos espacios en estructuras más adaptadas al objetivo, como por ejemplo árboles[1] o autómatas (grafos, en general) [Bahl93c], [Gopalakrishanan95], [Li95]; el uso de arquitecturas no integradas (como se comentó más arriba), en las que etapas de menor demanda computacional (por ejemplo usando técnicas de *fast-match* [Bahl92] [Bahl93c] [Labute95]), disminuyen el espacio de búsqueda sobre el que posteriormente trabajarán algoritmos más costosos; y, por supuesto, técnicas de búsqueda en haz (*beam-search*), combinadas con *look-ahead* y el uso de información lingüística en el proceso, en las que el objetivo es eliminar cuanto antes aquellas hipótesis poco susceptibles de formar parte de la solución final [Ney92] [Ortmans96] [Ortmans97b] [Steinbiss94], de modo que se consiguen reducciones muy considerables en el espacio de búsqueda, sin perjudicar notablemente la tasa de reconocimiento obtenida. En sí estas técnicas no serían *admisibles*, en el sentido de que no podemos asegurar la obtención del camino óptimo, al haber eliminado parte de las posibilidades, pero cuidadosos ajustes de los parámetros de control de los algoritmos de reducción, permiten dichos recortes manteniendo elevada la probabilidad de que el camino óptimo siga siendo activo.

También podríamos hablar de técnicas de organización algorítmica, que casi entrarían en el primero de los puntos descritos, y que se limitan a reorganizar el espacio de búsqueda de forma adecuada para limitar los recursos demandados, fundamentalmente memoria [Iwasaki97], o bien se basan en la generación dinámica e incremental del espacio de búsqueda [Sagerer96].

En nuestro caso, nos centraremos fundamentalmente en el último punto, en el que se analizarán en primer lugar las estructuras de árboles o grafos genéricos (con más o menos restricciones). Dichas técnicas serán aplicables para cualquiera de las arquitecturas estudiadas (integradas y no integradas), ya que en ambos casos hay módulos susceptibles de incorporarlas.

Centrándonos ya en arquitecturas no integradas, se va a abordar el estudio profundo y la ampliación de las ideas mostradas en [Ferreiros98b], en el que se describe la utilización de listas de preselección dinámicas (i.e., de tamaño variable), como entrada a un módulo de verificación. Para ello se utilizarán técnicas de estimación paramétricas y no paramétricas (i.e., asumiendo o no algunas suposiciones sobre la distribución a modelar), tomando decisiones de dicha longitud en función de parámetros disponibles en el reconocedor en cada momento (e.g. longitud de la palabra, coste asociado, longitud de la cadena reconocida, distribución de costes en los primeros candidatos, etc.). Entre las

1. Respecto al uso de árboles, hay que tener en cuenta que no siempre tienen por qué reducir espacio de búsqueda, sino producir el efecto contrario. Estamos pensando, por ejemplo, en lo que sucede al utilizar dichas estructuras combinadas con modelos de lenguaje en los que hay que replicarlas para mantener de forma adecuada las historias del proceso de búsqueda (aunque si además introducimos técnicas de búsqueda en haz (*beam-search*) se puede compensar el incremento de espacio producido, dando lugar a sistemas más rápidos).

técnicas posibles, se planteará inicialmente el uso de análisis de regresión para selección de características, al estilo de [Siu97], y redes neuronales.

Este tema de decisión de longitud de listas de preselección, enlaza directamente con la idea de *medidas de confianza* [Cole95] [Fetter96] [Siu97], y mecanismos de rechazo [Moreno90]. La mayoría de los sistemas de reconocimiento asignan valores de probabilidad (o coste) a cada una de las hipótesis, de modo que de la ordenación en función de dichos valores se obtiene una lista de candidatos. Lo que se planteará es el estudio de mecanismos específicos de rechazo, si la fiabilidad de una hipótesis (estimada por algún mecanismo concreto, a partir, como siempre, de parámetros disponibles durante el proceso), cae por debajo de un cierto umbral. Si los algoritmos de estimación de longitudes de listas de preselección funcionan correctamente, a una lista más larga le corresponderá una menor fiabilidad en los candidatos propuestos, con lo que podrán extrapolarse condiciones sobre este particular, como base de algoritmos de rechazo, llegando a unificar la formulación de ambos aspectos.

2.4 Alfabetos y diccionarios

Entenderemos por alfabeto el conjunto de unidades utilizadas para modelar el habla. Así, tendremos un modelo acústico para cada uno de los elementos del alfabeto. De entrada, no haremos ninguna asunción sobre la entidad de las mismas.

Igualmente, al referirnos a diccionarios, lo haremos en un sentido ligeramente más amplio que el habitual, añadiendo a la lista de palabras válidas en nuestro entorno o aplicación, la representación segmental de cada una de ellas a partir de las unidades que conforman el alfabeto. De nuevo, no haremos ninguna restricción al respecto, con lo que será perfectamente válido (y de hecho, es uno de los objetivos perseguidos), que haya más de una representación para una misma palabra dentro del diccionario, utilizando estructuras de datos adecuadas.

La definición (entendida como resolución) de dichas unidades varía significativamente en función de los requisitos impuestos por la tarea. Así, podemos tener un único modelo para cada una de las palabras del diccionario (i.e., alfabeto y diccionario coinciden en número de entradas), en aplicaciones sencillas de pequeño vocabulario [Rabiner88]. En el otro extremo tenemos la propuesta de *fenones* [Bahl93b] en la que sugiere un modelo mucho más pequeño, acercándose en longitud a la de una trama de la señal de voz.

Independientemente del tipo de unidades seleccionadas, el repertorio de las mismas debería cumplir, idealmente, una serie de propiedades [Holter98]:

1. *Consistencia*, es decir, que diferentes realizaciones de la misma unidad, deberán tener características similares.
2. *Entrenabilidad*, de modo que en nuestra base de datos de entrenamiento dispongamos de un número suficientemente elevado de ejemplos de cada unidad, como para conseguir modelos acústicos fiables.
3. *Economía*, tratándose en este caso de mantener su número dentro de unos ciertos límites, para evitar problemas como el aumento de carga computacional, o la dificultad de entrenamiento
4. *Cobertura*, de modo que las unidades seleccionadas cubran razonablemente todos los eventos acústicos posibles
5. *Concatenabilidad*, con la idea de que las palabras puedan ser descompuestas como una secuencia de dichas unidades

De todas ellas, es fundamental atender al binomio consistencia-entrenabilidad. Así, a medida que elegimos unidades de mayor longitud, incrementamos su consistencia, pero dificultamos la tarea de entrenamiento, al disponer de un menor número de ejemplos con los que llevar a cabo el entrenamiento. Por el contrario, unidades de menor longitud son menos consistentes, pero mucho más fácilmente entrenables, dado el mismo conjunto de entrenamiento.

Tradicionalmente se han considerado unidades basadas en criterios fonéticos, de modo que el repertorio ha sido seleccionado por expertos lingüistas. En la literatura podemos encontrar múltiples variaciones de esta aproximación: modelos de alófonos [Bahl89] con dependencia o independencia del contexto [Schwartz85], dependientes de la palabra, difonemas, trifonemas [Lee88], sílabas, demisílabas, etc. El problema en estos casos es la falta de adecuación entre las unidades definidas lingüísticamente y el contenido acústico real disponible en las bases de datos [Rossi94] [Holter98]. También es de destacar la consideración de una unidad propuesta recientemente, el semifonema (*demiphone*, en la nomenclatura inglesa) que ha demostrado conseguir buenos resultados, especialmente cuando los datos de entrenamiento son escasos [Mariño00].

Como intento de solución al problema, han surgido estrategias que proponen el uso de unidades subléxicas, generadas automáticamente (en principio) a partir del análisis acústico de la señal de voz.

La problemática en este caso es, por supuesto, la creación de un alfabeto consistente, y el diseño de los mecanismos que permitan determinar un conjunto óptimo de unidades, de acuerdo con algún criterio objetivo. En esta línea podemos citar los trabajos de Bahl [Bahl93b], y su propuesta de los fenones, como unidades generadas en un proceso de cuantificación vectorial, y modeladas con HMMs extremadamente sencillos. Por otro lado, también es de destacar la propuesta de unidades basadas en segmentos acústicos (*acoustic segment unit (ASU)*). La aproximación más típica a esta idea utiliza técnicas de segmentación y aglomeramiento basados en criterios objetivos [Ostendorf96].

Adicionalmente, surge el problema de la falta de conocimiento lingüístico incorporado al proceso. En cualquier caso, es el precio a pagar por la adopción de una estrategia de diseño consistente a lo largo del reconocedor[1]. Igualmente, uno de los problemas fundamentales en este caso es la generación de las pronunciaciones para las palabras del diccionario, ya que no hay correspondencia directa entre las unidades y las palabras. En nuestro caso, partiremos de un enfoque híbrido, en el que la selección inicial del repertorio de unidades se hará de acuerdo a criterios lingüísticos, y, a continuación, se plantearán mecanismos automáticos dirigidos por datos para el aglomeramiento de dichas unidades en función del análisis explícito de la base de datos [Rabiner89a][Macías96], en la misma línea que las ideas de agrupamiento de trifonemas descritas descritas en [Rabiner89b], [Lee88,Lee89] o [Fissore91].

Enlazando con la característica de *concatenabilidad*, entramos en la problemática de la selección de la descomposición en unidades de cada palabra del diccionario, que no es más que la selección de la *pronunciación* de aquella. Si los criterios que han generado el alfabeto de símbolos son lingüísticos, posiblemente habrá un mecanismo de generación automática o semiautomática de dicha descomposición. En caso contrario, y como se comentó más arriba, será imprescindible abordar mecanismos especiales. Este aspecto adicional de la variabilidad de la señal de voz, en lo que se refiere a las pronunciaciones alternativas que podemos encontrar, incluso para un mismo hablante, inevitablemente lleva a errores de reconocimiento cuando únicamente se contempla la pronunciación canónica a la hora de acceder al diccionario [Wooters94], ya que puede darse el caso de que la pronunciación de facto tenga poco o nada que ver con aquella.

La aproximación tradicional a ambos problemas (selección de unidades y de pronunciación), se ha basado en módulos importados del campo de la conversión de texto a voz. El enfoque más sencillo es utilizar como alfabeto a modelar, la lista de unidades propuestas por un sistema de conversión de texto a voz, y la representación alofónica de cada palabra en el diccionario es única, aquella considerada como estándar, bien derivada de un conversor grafema-alófono, o bien extraída de un diccionario en el que las formas fonéticas han sido generadas manual o semi-automáticamente [Ferreiros98a].

La preocupación por introducir modelado explícito de variaciones de pronunciación surgió hace más de 25 años en el "*IEEE Symposium on Speech Recognition*" y el interés ha resurgido con fuerza en los últimos años, fuertemente motivado por el carácter cada vez más realista de las bases de

1. Cuando decimos *consistente* nos referimos a que se usan criterios comunes y objetivos para estimar *todos* los elementos que intervienen en el reconocedor, desde las unidades elementales a entrenar, hasta los modelos de Markov asociados a las mismas; es decir: todo se entrena.

datos con las que se cuenta [Strik99]. En la parte de modelado del diccionario (pronunciación), podemos describir sistemas que aplican reglas heurísticas para la generación de múltiples entradas, y que han mostrado conseguir mejores resultados que los que únicamente usan la pronunciación canónica [Westendorf96] [Schmid93] [Aubert95].

Un prometedor enfoque en la actualidad se basa en utilizar criterios objetivos para la estimación de ambas fuentes de conocimiento. En concreto, en [Holter98] se hace una interesante descripción de técnicas centradas en la generación de múltiples pronunciaciones, usando técnicas basadas en datos, y aplicando el criterio de máxima verosimilitud.

Abundando más en el tema, pero desde la perspectiva de sistemas basados en reglas en nuestro caso, partiremos en la estrategia mixta de utilización de un conversor grafema-fonema estándar descrita en [Ferreiros98a], aplicando reglas de pronunciación alternativas. De entrada se utilizarán reglas de variación comúnmente aceptadas como estándar en la comunidad castellano-hablante, y se abordará el estudio de modificaciones dialectales específicas, además de enlazar con técnicas de guiado por datos para la generación de grafos de pronunciación y validación posterior [Holter98], basándonos en producciones reales de las palabras [Elvira97].

Adicionalmente, se va a plantear el estudio detallado de los efectos de la variación del diccionario, para obtener conclusiones fundamentadas sobre la condición de dependencia o independencia del vocabulario [Hon89] [Hong90] [Villarrubia96], intentando aislar aquellos factores que pudieran enmascarar variaciones en la tasa de reconocimiento (e.g. la longitud media de las palabras del diccionario).

2.5 Modelos de lenguaje

La tarea de un modelo de lenguaje es capturar las restricciones que existen a la hora de combinar palabras para generar las frases posibles en un lenguaje dado. Dichas restricciones, de carácter sintáctico, semántico y pragmático, son difícilmente integrables como fuente de conocimiento en un SRAH. La importancia de dichos modelos es que permiten guiar de forma más eficaz a los reconocedores en el espacio de búsqueda sobre el que se mueven, si bien también pueden utilizarse para corregir a posteriori la salida de los mismos.

Sin embargo, en la literatura hay multitud de ejemplos de dicha integración, desde la utilización de diversos tipos de formalismos sintácticos y semánticos hasta las gramáticas probabilísticas. En el primer caso, debido a problemas de cobertura y de complejidad computacional, su incorporación a los SRAH no ha sido ni mucho menos inmediata, y únicamente se han usado formalismos relativamente simples, basados en gramáticas regulares y de contexto libre, por ejemplo.

En sistemas de gran vocabulario, los métodos probabilísticos han sido los más utilizados [Jelinek91] [Cerf92], fundamentalmente por su adecuación al entrenamiento automático, utilizando grandes bases de datos etiquetadas convenientemente. Igualmente, el uso de técnicas específicas de suavizado les dotan de una gran robustez, habiéndose propuesto modelos basados en categorías [Paeseler89] [Lee89b], en lugar de palabras, perdiendo por tanto potencia en reducción de espacio de búsqueda, pero incrementando su generalidad y su facilidad de entrenamiento, aunque permanece sin solución definitiva el repertorio de categorías a utilizar. Alternativas a esto lo constituyen mecanismos de estimación automática de dicho repertorio, aunque tampoco hay conclusiones claras sobre los criterios a utilizar, que generalmente se basan en disminución de la perplejidad, no habiéndose demostrado la relación directa entre este parámetro y la tasa de reconocimiento obtenida, dándose el caso de gramáticas con menor perplejidad que otras que, sin embargo, consiguen mejorar las tasas de reconocimiento finales. En toda esta discusión, hay que tener siempre presente el compromiso a establecer con la cobertura real que deseamos tener.

No abundaremos más en este tema, por no ser objetivo de esta tesis profundizar en el mismo, de modo que nos limitaremos a remitirnos al detallado estudio presentado en [Jones94]. La justificación de introducir aquí este tema es la necesidad de matizar cuantitativamente todos los resultados obtenidos

en esta tesis aplicando el modelo de lenguaje adecuado si planteamos su uso en tareas en las que se procese habla continua o, incluso cuando se trate de habla aislada[1].

2.6 Técnicas de entrenamiento

Los sistemas de entrenamiento utilizados en SRAH tienen una importancia capital, al ser la base de estimación de los parámetros de los modelos acústicos y lingüísticos utilizados, modelos que se usarán en el proceso de reconocimiento [Rabiner89a].

Tradicionalmente, y centrándonos ya en reconocedores basados en HMM, el criterio más extendido para la estimación de estos ha sido el de máxima verosimilitud [Bahl83], y es ahí donde radica uno de los principales defectos de esa formulación: El criterio de máxima verosimilitud trabaja en el sentido de optimizar los parámetros de una distribución determinada en función de unos datos disponibles (observados), pero el rendimiento de un SRAH se mide normalmente por su tasa de reconocimiento estimada. Así, no hay conexión directa entre el criterio de estimación usado para generar los HMM y la función objetivo final que queremos maximizar.

Un efecto lateral de este planteamiento es la ausencia de criterios de discriminación (como se comentó más arriba) en el proceso de entrenamiento, con lo que los modelos no contienen en sí dicha propiedad, y es ahí donde, como se comentó anteriormente, las redes neuronales muestran su potencia.

En este estado de cosas, durante los últimos años, se han venido desarrollando una serie de ideas orientadas a solucionar este problema y dotar de capacidad discriminadora explícita a los HMM. Por ejemplo, podemos citar técnicas basadas en entrenamiento correctivo (*corrective training*) [Bahl93b], estimación de máxima información mutua (*Maximum Mutual Information Estimation MMIE*) [Bahl86] [Valtchev97] y métodos basados en error de clasificación mínimo (*mimimum clasification error MCE*) y descenso generalizado probabilístico (*Generalized Probabilistic Descent GPD*) [Bahl93a] [Juang97]. En todos los casos lo que se modifica es la función objetivo a maximizar, de acuerdo a criterios directamente relacionados con la capacidad de discriminación o el error de clasificación.

En lo que respecta a la relación con las arquitecturas, en la literatura hay intentos de aplicar técnicas de entrenamiento dependiente y conjunto [Chiang94] [Chiang96], en aquellos sistemas multi-módulo disponibles, en los que, en general, se hace independientemente.

En esta tesis no abordaremos la incorporación de estos métodos, quedando planteados para las líneas futuras.

2.7 Validación estadística y medidas de rendimiento

En el año 1995, se completó un informe sobre el estado del arte en tecnología del lenguaje humano [Cole95], y en ella se identificaba la evaluación como un aspecto crucial en procesamiento de habla y lenguaje natural. De hecho, el programa DARPA ha invertido una gran cantidad de recursos en las evaluaciones periódicas de los sistemas generados por laboratorios en todo el mundo. Recientemente el grupo de trabajo EAGLES ha editado una serie de informes centrados en la definición de estándares y recursos para sistemas de lenguaje hablado [Gibbon98]. El tema está aún poco asentado y la definición de estándares y evaluaciones en este área es visto con cierto excepticismo por parte de la comunidad investigadora. Sin embargo, hay multitud de pautas y recomendaciones para guiarnos en nuestro caso, con lo que nuestra intención será dar unas ligeras pinceladas que den más fundamento a las medidas y las conclusiones sobre la bondad de tal o cual alternativa.

1. En este caso estamos pensando en que las entradas del diccionario utilizado estarán sujetas a una determinada distribución de su frecuencia de uso en la tarea sobre la que se aplique, lo que modificará significativamente los resultados reales obtenibles si las bases de datos de evaluación no reflejan adecuadamente dicha distribución.

En el desarrollo y validación de algoritmos y sistemas de reconocimiento en general, es de fundamental importancia analizar hasta qué punto las diferencias de rendimiento observadas, medidas de acuerdo con ciertos criterios, son significativas estadísticamente.

En todos los casos, la medida de rendimiento de un sistema vendrá relacionada directamente con el tamaño de la base de datos con que nos enfrentemos. Así, a mayor tamaño, mayor seguridad tendremos acerca de la fiabilidad de los resultados y de las diferencias que hayamos obtenido entre sistemas de distintas características.

En la literatura pueden encontrarse numerosas referencias a este tema, y, por citar algunas, nos centraremos en las que dan soporte a la estrategia que aplicaremos en nuestro caso.

En primer lugar mencionaremos los trabajos de Gillick y Cox [Gillick89], que no sólo solucionan la evaluación de la validez estadística de mejoras en cada algoritmo o técnica, sino que permiten evaluar el rendimiento comparativo de sistemas diferentes, junto con los de Cavanesio [Canavesio92] y Wu [Wu94]. En principio nos limitaremos a tratar el caso de sistemas de reconocimiento de palabras aisladas (para el caso de habla conectada, podemos referenciar también a [Gillick89], y para habla continua a [Pallet89]).

No entraremos en detalle en la formulación estadística de los algoritmos implicados, limitándonos a decir que las aproximaciones al problema suelen pasar por la estimación del número de errores que comete cada sistema, y las tasas de error obtenidas. Hay que tener en cuenta, además, que el hecho de usar o no la misma base de datos para hacer las comparaciones, impone restricciones en los métodos, que serán distintos para el caso de bases de datos dependientes e independientes. En el primer caso, es de destacar el test de McNemar (usado extensamente, por ejemplo, en los tests de evaluación de DARPA RM).

Un método adicional de validación es el expuesto en [Weiss93], en el que se calculan las bandas de fiabilidad resultantes de imponer una tasa determinada de confianza a dichas medidas. Así, por ejemplo, podremos fijar una tasa de confianza del 95%, y el método estimará los márgenes entre los que se podría mover la tasa de reconocimiento/error obtenida.

La aplicación de estas medidas *objetivas* es de vital importancia para poner en su justo sitio y con su justa medida la eficacia real de modificaciones y mejoras que, en ocasiones, se diseñan para solucionar problemas puntuales, pero no acaban de aportar soluciones globales y efectivas *en media*, o *estadísticamente significativas*.

En contraposición a lo indicado en el párrafo anterior, queremos señalar que, en ciertos casos, la consecución de mejoras marginales[1] puede ser importante. Estamos pensando en casos de locutores especialmente difíciles (por su características acústicas o de pronunciación) o producciones de voz con ruidos específicos observados raramente, para los que se pueden plantear soluciones que no ofrezcan diferencias con validez estadística para la población global pero sí solucionen el problema concreto planteado. Las mejoras marginales (o incluso ligeros empeoramientos globales) de dichas soluciones nunca se podrán apoyar sobre un conjunto elevado de datos que garanticen su fiabilidad, pero son imprescindibles en sistemas de acceso universal, por ejemplo servicios de suministro de información telefónica.

En cualquier caso, no debemos perder de vista una reflexión importante sobre este tema [Ferreiros96]: puede suceder que dicha validación estadística no se cumpla significativamente, pero se verifiquen mejoras apreciables en todos los casos de aplicación de una técnica determinada. Así, dichas técnicas “son ideas que se proponen con la cautela de no haber podido demostrar significativamente su validez, pero que no queremos abandonar a la espera de que sean útiles en los sistemas de reconocimiento y que queden validadas en futuras experimentaciones en otras aplicaciones o con más datos”.

1. Entendiendo por ellas aquellas que no son estadísticamente significativas o bien que afectan a un conjunto muy reducido de elementos de las bases de datos de evaluación.

3 Estudio de arquitecturas

3.1 Introducción

El objetivo de este capítulo es estudiar y evaluar la capacidad de cada una de las arquitecturas planteadas de cara a enfrentarse a tareas de distinta complejidad y entorno de aplicación. El objetivo último es el estudio de métodos y estrategias de diseño en reconocedores multi-módulo, planteándonos una evaluación cualitativa y cuantitativa referida al compromiso tasa de reconocimiento vs. tiempo de proceso.

Partiremos de unas consideraciones iniciales sobre las estrategias integradas y no integradas, como paso previo a la descripción de las arquitecturas evaluadas en el contexto de esta tesis. El grueso del capítulo lo constituirá la discusión sobre arquitecturas multi-módulo y la propuesta de una metodología de diseño de las mismas, así como la evaluación experimental realizada. Un apartado de conclusiones finales cierra el capítulo.

3.2 Consideraciones sobre arquitecturas integradas vs. no integradas

En el Apartado 2.2 del encuadre científico-tecnológico, al referirnos a las alternativas arquitecturales, se plantearon dos divisiones ortogonales de sistemas de reconocimiento de habla. La primera hacía referencia a los paradigmas de hipótesis-verificación o de verificación en un sólo paso, y la segunda, a los enfoques integrados y no integrados.

Tras esa discusión, queda claro que nuestro enfoque al hablar de sistemas no integrados parte de la separación entre los módulos que hacen uso de fuentes de información acústica y los que usan fuentes de información léxica (que podríamos generalizar, ampliando el rango a fuentes de conocimiento sintáctico, semántico e, incluso, pragmático). Si planteamos esquemas en varios niveles de detalle que utilicen progresivamente fuentes de conocimiento cada vez más refinadas, la frontera entre sistemas integrados y no integrados sigue estando clara aunque se haga un uso simultáneo de distintas fuentes de información en algunos módulos (arquitecturas integradas en sí) conectados entre sí, usando información cada vez más refinada en el proceso (arquitecturas no integradas en el sistema conjunto), proporcionando cada uno de ellos información más depurada a los siguientes.

En los apartados de este capítulo haremos referencia estricta a la clasificación planteada en el encuadre, considerando el uso de sistemas (módulos individuales) integrados o no integrados enmarcados en una arquitectura final no integrada, al estar basada en el paradigma hipótesis-verificación.

3.3 Arquitecturas evaluadas

Como discutimos en el Apartado 2.2, nuestro interés fundamental será establecer la validez y aplicabilidad de distintas arquitecturas, fundamentalmente las que usan sistemas basados en una estrategia integrada o no integrada, con la posibilidad adicional de realizar el proceso en una o varias etapas.

Obviamente, cualquier estudio en este terreno implica el diseño e implementación de un sistema con tales características. Así, en este apartado describiremos los sistemas desarrollados pertenecientes a cada enfoque arquitectural, sistemas que serán susceptibles de ser combinados para obtener alternativas más potentes.

3.3.1 Sistemas integrados

En el desarrollo de la tesis se han diseñado e implementado dos sistemas que usan el enfoque integrado, al combinar en un único paso la información acústica y léxica disponible (como se muestra en la Figura 3-1):

- Reconocedor basado en el algoritmo de Viterbi sobre un diccionario lineal
- Reconocedor basado en el algoritmo de un paso sobre un diccionario en forma de árbol

Evidentemente, ambos sistemas producen los mismos resultados de salida, en cuanto a palabras reconocidas y tasas de reconocimiento obtenidas, pero el segundo tiene como ventaja fundamental el considerable ahorro de tiempo fruto de la compresión del espacio de búsqueda, al compartir los nodos iniciales del mismo para todas las palabras, como se discutirá en el Apartado 4.3.5 "Consideraciones de ahorro en tiempo de proceso", a partir de la página 84[1].

En los dos sistemas propuestos, partimos de la misma fuente de información léxica, un diccionario (organizado en forma lineal o en árbol) que guía el proceso de búsqueda acústico, no permitiéndose, por tanto, la generación de soluciones no directamente pertenecientes al diccionario

En estos sistemas se pueden utilizar distintos diccionarios (descritos en el Anexo B para cada una de las bases de datos y tareas en estudio) y, por supuesto, todas las alternativas de modelado que se discuten en el Apartado 5.1 "Selección de unidades" (modelado discreto y semicontinuo, con modelos dependientes o independientes del contexto, para distintos alfabetos).

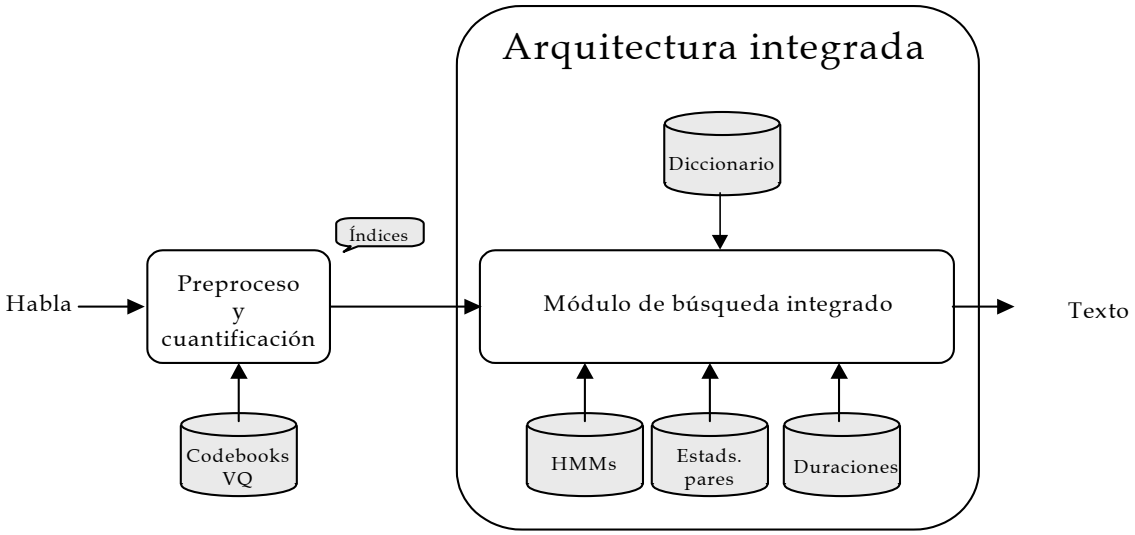

Figura 3-1: Arquitectura integrada

3.3.2 Sistema no integrado

Nuestro sistema no integrado consta de dos módulos fundamentales (como puede verse en la Figura 3-2, prescindiendo de los detalles referidos al módulo de preproceso y cuantificación): el primero encargado de generar una cadena o malla de unidades elementales (usando la información acústica disponible en un repertorio de modelos de mayor o menor precisión) y el segundo encargado de comparar dicha cadena o malla con las alternativas válidas pertenecientes al diccionario elegido del dominio de la aplicación (usando la información léxico-fonética disponible como descomposición de las palabras del vocabulario en unidades elementales).

1. Recordamos la consideración, ya hecha en el encuadre científico-tecnológico, al respecto de la posibilidad de que el espacio de búsqueda se incremente si usamos ciertos modelos de lenguaje. Nos referimos en este apartado al uso en reconocedores sin intervención modelado lingüístico.

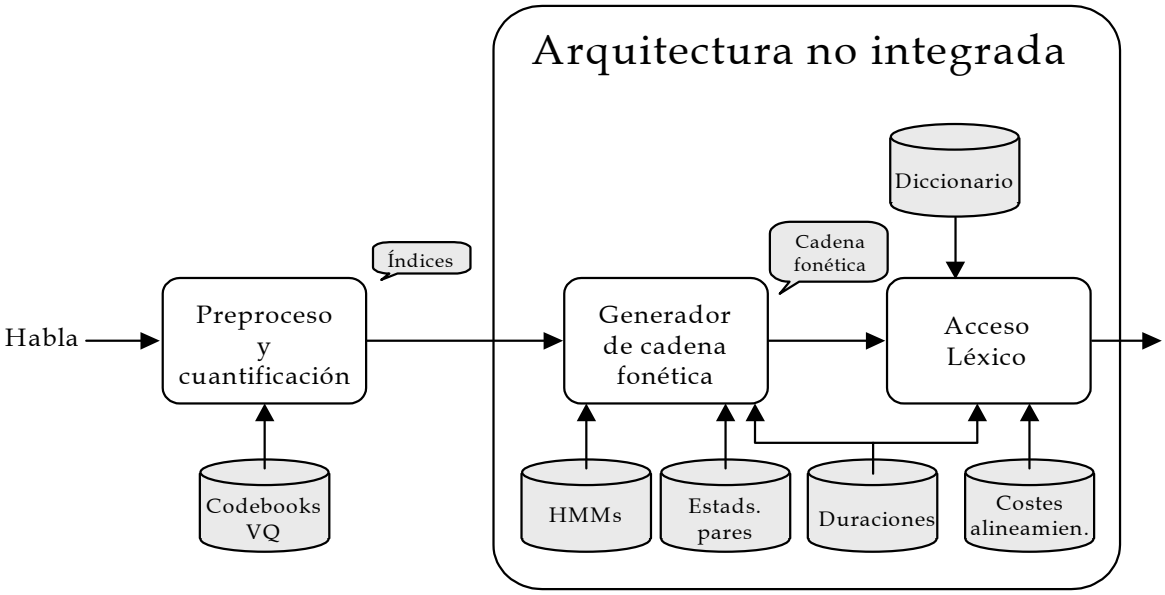

Figura 3-2: Arquitectura no integrada

La gran diferencia con el enfoque integrado es la falta de un guiado o restricción del espacio de búsqueda durante el proceso de búsqueda acústica (es decir, se usa la información acústica y la del diccionario en instantes distintos), lo que repercute, como ya se ha discutido, en una menor demanda computacional a costa de un previsible incremento en la tasa de error obtenida[1].

3.3.2.1 Módulo de generación de cadena fonética

Basado en el algoritmo de un paso, cuya versión básica para sistemas que usaban DTW ("Dynamic Time Warping"), puede encontrarse en [Bridle82] y [Ney84]. La versión que hemos utilizado en nuestro caso funciona síncronamente en el tiempo, lo que la hace apropiada para su implementación en tiempo real, y utiliza HMM's, como ya se ha comentado. Al algoritmo básico se le ha dotado de una serie de mecanismos adicionales:

- Integración de factores de modelado de duraciones en el proceso de búsqueda.
- Aplicación del proceso de reducción o no[2].
- Uso de estadísticas de pares de unidades, modificando la decisión de salto que se analiza en el algoritmo, restringiendo dicha posibilidad a aquellas unidades que puedan ser predecesoras de la que se considere en cada momento.
- Uso de modelos de silencio, lo que permite imponer o no el encadenamiento de modelos de silencio (inicial y/o final) al principio y/o final de la secuencia de símbolos. Dicha modificación es fundamental para hacer frente a los inevitables defectos en el proceso de detección de principio y fin.
- Generación de mallas (*lattices*) de unidades, en lugar de simples cadenas, lo que permite incrementar las alternativas de alineamiento en el acceso léxico, lo que puede ser contraproducente si la generación es excesiva.

1. Que habrá que evaluar en cada caso, discutiendo sobre el compromiso razonable entre ambos valores: coste vs. tasa.
2. Con *reducción* nos referimos a la posibilidad de tomar decisiones sobre el mejor estado en cada instante, trama a trama, dejando que únicamente dicho estado pueda encadenarse con los posibles siguientes. En caso de no usar gramáticas, no existe diferencia entre ese enfoque y el hecho de no usarlo, pero su uso es crucial en caso contrario, pudiendo producir resultados totalmente erróneos.

3.3.2.2 Módulo de acceso léxico

El hecho de estar diseñando un sistema no integrado, obliga a la inclusión de un módulo adicional *de acceso léxico*, que se encargue de *corregir* en lo posible los errores cometidos por el módulo acústico (por falta de guiado léxico) y que permita obtener un rendimiento lo más óptimo posible.

La idea es aprovechar en lo posible toda la información disponible, como por ejemplo, estadísticas de los errores cometidos, matriz de confusión entre unidades, duraciones de fonemas o de palabras, y utilizarla en un módulo de post-proceso a la salida del decodificador acústico, aunque es posible que el sistema trabaje síncronamente en el tiempo (usando técnicas de *parcial traceback*, como las aplicadas en [Macías92]), abriendo las puertas a sistemas más complejos en tiempo real (reconocimiento de habla continua, por ejemplo).

En nuestro caso particular, utilizamos una implementación de la opción basada en la que se presenta en [Fissore89], junto con modificaciones que aumentan el rendimiento del algoritmo [Macías92]. Se trata de un algoritmo de programación dinámica modificado para trabajar en un espacio tridimensional en el caso de que se traten *mallas* (*lattices*) de unidades en lugar de *cadenas* (*strings*). En el proceso de alineamiento se utilizan costes entrenados de substitución, inserción y borrado (siendo los dos primeros contextuales).

El módulo recibe a su entrada una cadena o malla de unidades fonéticas (que son la salida del decodificador acústico), y genera a su salida una serie de palabras candidato que el sistema clasifica como las que más probablemente representan a la cadena (malla) de entrada.

El espacio de búsqueda se organiza en forma lineal o de árbol, al igual que discutíamos en el caso de los sistemas integrados. Los resultados obtenidos son idénticos en ambos casos, salvo por la menor demanda computacional del segundo. Además, en el Apartado 4.3 "Estrategias de exploración/ búsqueda", se discutirá el uso de grafos de distintos tipos para la estructuración del espacio de búsqueda en este tipo de sistemas, como alternativa a los tradicionales.

3.3.3 Sistemas basados en hipótesis verificación

Como se discutió en el encuadre, los sistemas basados en el paradigma hipótesis-verificación tratan de obtener progresivamente versiones más refinadas de la producción de habla realizada. Su característica fundamental es ir reduciendo el espacio de búsqueda en cada caso, aplicando métodos y técnicas cada vez más complejos. Así, se invierte cada vez más esfuerzo computacional, pero sobre un espacio de alternativas más pequeño, de modo que disminuyamos el tiempo de proceso total, manteniendo las tasas de reconocimiento.

De este modo, es planteable usar cualquier combinación de los módulos o sistemas individuales vistos anteriormente. Más adelante haremos las consideraciones oportunas sobre compromisos entre tiempo de proceso y fiabilidad, pero resumiendo indicaremos aquí que se trata de encontrar la combinación óptima de etapas y de tamaños de lista de hipótesis (preselección) que maximice la tasa de reconocimiento obtenida, minimizando la demanda computacional.

La estrategia de hipótesis-verificación es extendible a sistemas multi-módulo, como se discutirá más adelante, en la que cada uno de los incluidos en la cadena funciona para el siguiente como un generador de hipótesis (módulo de preselección).

3.3.3.1 Módulo de hipótesis

A lo largo del desarrollo de esta tesis se hizo una experimentación bastante exhaustiva con los módulos desarrollados o los que se encontraban disponibles en nuestro Grupo [Macías94b]. A partir de ella, se optó por utilizar el sistema no integrado descrito en el Apartado 3.3.2, con distintas alternativas de modelado. Este sistema supone un compromiso razonable entre coste computacional y tasas obtenidas.

3.3.3.2 Módulo de verificación

Los módulos de verificación utilizados fueron en todos los casos los sistemas integrados descritos en el Apartado 3.3.1, al ser necesario en estos casos aplicar toda la potencia de clasificación disponible.

3.4 Consideraciones en sistemas multi-módulo

Cuando trabajamos con sistemas multi-módulo, entendiendo como tales aquellos que siguen el paradigma hipótesis-verificación, extendidos en general a más de dos módulos, hay que tener especial cuidado en considerar el compromiso entre distintos factores:

- Tiempo de proceso de cada módulo, medible en general normalizado por el número de palabras de las que se compone el diccionario[1], para permitir la generalización de la argumentación a tareas de distinto vocabulario
- Tasa de error obtenida, considerando un número dado de alternativas. En el último módulo de la cadena sólo nos importa la respuesta ofrecida en *primera posición*, pero en módulos intermedios necesitamos obtener valores de tasa en función del número de opciones que entregaremos al siguiente.
- Número de opciones que cada módulo ofrecerá al siguiente (dicho número definirá el tamaño del vocabulario activo). Este factor está estrechamente ligado al anterior y ambos fijan la calidad de la arquitectura final. Como veremos posteriormente, dicha calidad, medida como tasa de error final, no es simplemente el resultado de multiplicar las tasas de los sistemas en cascada, sino que hay una dependencia más sutil.

La discusión que vamos a establecer la haremos sobre la suposición de que tratamos de sistemas de reconocimiento de habla aislada.

3.4.1 Esquema, nomenclatura y definiciones

En la Figura 3-3 se muestra la estructura genérica de una arquitectura de reconocimiento multi-módulo.

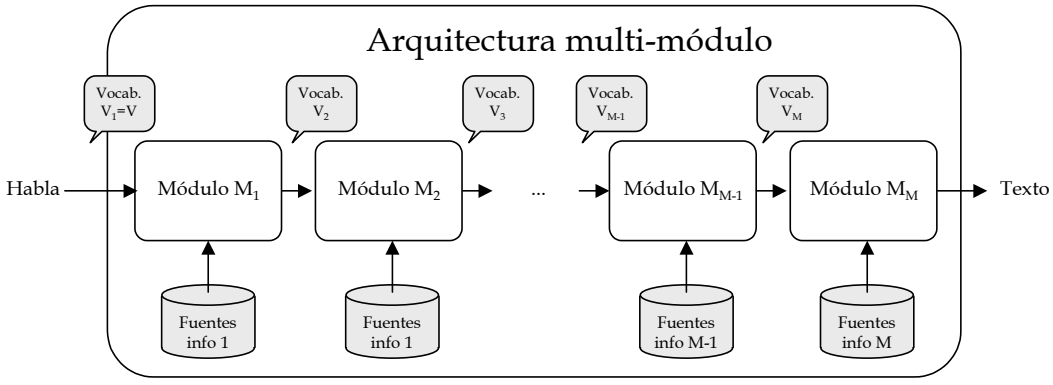

Figura 3-3: Arquitectura multi-módulo

Sobre ella, estableceremos las siguientes definiciones y nomenclatura:

1. Aunque puede darse el caso, la dependencia del tiempo de proceso no tiene por qué ser linealmente dependiente del número de palabras directamente (puede serlo del número de nodos del grafo, del número de nodos finales, etc., como se discutirá en el Apartado 4.3.5 a partir de la página 84 donde se hacen consideraciones sobre el tiempo de proceso en función de la estructura del espacio de búsqueda usado).

M_i Módulo *i* de la cadena, con *i = 1..M*

V Tamaño del vocabulario usado en la tarea

V_i Vocabulario activo considerado por el módulo M_i de la cadena, con *i = 1..M*, $V_i \leq V$ (como caso particular, $V_1 = V$)

$\psi_i(\lambda)$ Tasa de reconocimiento (inclusión) obtenida por el módulo M_i en la posición λ, con *i = 1..M* al enfrentarse al diccionario completo V

$\psi_{ij}(\lambda, V_i, \ldots, V_j)$ Tasa de reconocimiento (inclusión) obtenida por la combinación de la secuencia de módulos $M_i \ldots M_j$ en la posición λ, con *i = 1..M-1*, *j = 1..M* y *j > i*

$\psi_i'(\lambda, V_i)$ Tasa de reconocimiento (inclusión) obtenida por el módulo M_i en la posición λ, si la palabra objetivo está incluida en el vocabulario activo V_i, con *i = 1..M-1* y $\lambda = 1..V_i$

Γ Tiempo medio total de proceso empleado por la arquitectura para procesar una producción de habla

τ_i Tiempo medio de proceso por palabra (producción de habla) y por entrada del diccionario en el vocabulario dado en el módulo M_i, con *i = 1..M*

3.4.2 Tiempo de proceso

A partir del esquema de la Figura 3-3 y de las definiciones vistas en el apartado anterior, es fácil establecer que el tiempo medio de proceso total empleado por la arquitectura para cada producción de habla puede aproximarse por[1]:

$$\Gamma = \sum_{i=1}^{M} \tau_i \cdot V_i$$

A medida que avanzamos en la cadena de módulos, el tiempo medio de proceso τ_i aumenta (dado que usamos técnicas de modelado y búsqueda más potentes y complejas) y el tamaño del vocabulario activo V_i disminuye.

El objetivo fundamental de una arquitectura multi-módulo es conseguir tasas altas de reconocimiento con esfuerzos computacionales lo más bajos posibles, más bajos que los que harían falta si usáramos el último módulo de la cadena presentándole el vocabulario V completo (que además nos entregará la máxima tasa posible), es decir:

$$\Gamma = \sum_{i=1}^{M} \tau_i \cdot V_i < \tau_M \cdot V$$

Por supuesto, asumimos que debemos mantener la tasa de reconocimiento obtenida lo más cercana posible a la del sistema de verificación enfrentado al diccionario completo: es planteable perder algo de tasa, teniendo en cuenta el compromiso entre dicha pérdida y el coste computacional implicado. Llegados a este punto, la visión es doble: por un lado nos interesa saber si efectivamente interesa la combinación de M módulos frente a la de un único paso y, por otro, querremos evaluar la pérdida de tasa en función de las restricciones en cuanto a carga computacional que decidamos imponer a nuestro sistema.

1. Hablamos de aproximación porque asumimos en todos los casos valores medios. Un cálculo exacto tendría que tener en cuenta la aportación de coste computacional de cada palabra individualmente.

Si particularizamos al caso básico de dos módulos (usando ahora los subíndices $_{hipot}$ y $_{verif}$ para referirnos a los módulos de hipótesis y verificación respectivamente, tendremos:

$$\Gamma = \tau_{hipot} \cdot V + \tau_{verif} \cdot V_{verif} < \tau_{verif} \cdot V$$

En la Tabla 3-1 se muestra un ejemplo de los tiempos medios de proceso por palabra para distintos módulos de proceso obtenidos para tres experimentos diferentes sobre la tarea VESTEL-L[1] (con los diccionarios 1952, 5000-85-15 y 10000-85-15[2]) usando modelado semicontinuo independiente del contexto con el alfabeto `alf45`[3] en el módulo de preselección (generación de cadena fonética seguida de acceso léxico); y modelado semicontinuo dependiente del contexto, también con `alf45`, usando 800 distribuciones finales, en el módulo de verificación. En dicha tabla se incluyen los valores en milisegundos. Las condiciones de tiempo real para nuestro experimento equivaldrían a un tiempo de proceso total igual a la duración media de las palabras de nuestra base de datos: 600 milisegundos en este caso.

Tabla 3-1: Tiempos de proceso medios por palabra

	Hipótesis	**Verificación**	**Veces[1]**
Tarea	$\tau_{hipot} \cdot V$	$\tau_{verif} \cdot V$	
1952	*76'7 ms*	*1093 ms*	*14*
5000-85-15	*119'4 ms*	*2622 ms*	*22*
10000-85-15	*189'1 ms*	*5383 ms*	*28*

1. Número de veces que el módulo de preselección es más rápido que el de verificación.

Como puede observarse, los tiempos son razonables en el caso de 1952 palabras (comparando con las condiciones de tiempo real), incluso para el módulo de verificación. Sin embargo, al subir el tamaño de los diccionarios a procesar, el tiempo invertido en el módulo de verificación se hace excesivo.

La expresión que nos da el ahorro relativo de tiempo (A_t) entre el uso de la arquitectura basada en hipótesis-verificación frente a la de verificación sería:

$$A_t(\%) = 100 \cdot \frac{\tau_{verif} \cdot V - (\tau_{hipot} \cdot V + \tau_{verif} \cdot V_{verif})}{\tau_{verif} \cdot V} = 100 \cdot \left(1 - \frac{\tau_{hipot}}{\tau_{verif}}\right) - 100 \cdot \frac{V_{verif}}{V}$$

donde puede verse que el ahorro relativo depende de dos factores:

- Uno dependiente de la relación de tiempos medio de proceso para cada entrada del diccionario entre los módulos de hipótesis y verificación :

$$100 \cdot \left(1 - \frac{\tau_{hipot}}{\tau_{verif}}\right)$$

- Otro dado por la relación entre los tamaños de la lista de preselección y el vocabulario completo:

1. Base de datos realista sobre línea telefónica, descrita en detalle en el Anexo B.2 a partir de la página 189, que permite la evaluación sobre casi 10000 producciones de habla aislada. El sufijo -L indica que se usó la técnica del *Leave-one-out* para incrementar la fiabilidad estadística de los resultados.
2. Diccionarios de 1952, 5000 y 10000 palabras, para la tarea VESTEL-L, cuya construcción se describen en detalle en el Anexo B.2.3, a partir de la página 190.
3. Compuesto por 45 unidades diferentes y que se describe en detalle en el Anexo D.2.2, a partir de la página 204.

$$100 \cdot \frac{V_{verif}}{V} = \texttt{\%LongitudPreselección} = \%\ \text{(Long. lista preselección /Tamaño Vocabulario)}$$

donde `%LongitudPreselección` es el tamaño de la longitud de la lista de preselección calculado como porcentaje del tamaño del diccionario que, como se verá, suele ser el eje que usamos como referencia en las gráficas de tasa de error de inclusión de este tipo de sistemas[1].

Es de destacar que el ahorro de tiempo puede ser negativo, con lo que la combinación de sistemas puede ser contraproducente. Sin embargo, las condiciones habituales de los módulos de hipótesis y verificación son tales que hacen prácticamente imposible que sea así. En general, se cumplirá que $\tau_{verif} \gg \tau_{hipot}$, con lo que:

$$A_t(\%) \cong 100 - \texttt{\%LongitudPreselección}$$

lo que justifica los ahorros tan sustanciales de tiempo que pueden conseguirse con esta estrategia. Incluso en casos extremos es así. Pensemos por ejemplo en un sistema en el que $\tau_{verif} = 2 \cdot \tau_{hipot}$, tendríamos $A_t(\%) = 50 - \texttt{\%LongitudPreselección}$, donde si pensamos en una longitud de lista del 10% implicaría un ahorro de tiempo de un 40%, razonablemente alto y que podría servir perfectamente para nuestros propósitos.

En definitiva, la expresión general para el ahorro de tiempo vista más arriba nos da idea de que es posible conseguir ahorros muy importantes en tiempo de proceso para prácticamente todas las longitudes de preselección planteables y un amplio margen de relaciones de tiempo de proceso entre los módulos involucrados.

Obviamente, aún nos queda estudiar el impacto de la reducción de cómputo en la tasa de inclusión del sistema. En la Figura 3-4 se muestra el porcentaje relativo de pérdida en tasa de reconocimiento para el primer candidato (tomando como base la máxima obtenible por el módulo de verificación enfrentado al diccionario completo) en función del ahorro relativo de tiempo de proceso usado comparado con el uso de un sistema en un único paso, para los tres diccionarios vistos más arriba. La observación a partir del mismo es que es posible reducir drásticamente el tiempo de proceso manteniendo la tasa de reconocimiento en valores muy cercanos al máximo alcanzable, lo que vuelve a favorecer la preferencia de sistemas en varios pasos frente a uno único. Así por ejemplo, en las tres tareas es posible reducir el tiempo de proceso entre un 85% y un 95% con una pérdida relativa de tasa inferior al 1%.

Finalmente, en la Figura 3-5 se muestra la pérdida relativa de tasa obtenida en función de la fracción de tiempo real utilizada, para las tres tareas evaluadas (la gráfica superior usa un eje de abscisas logarítmico y la inferior uno lineal, siendo ésta última un aumento detallado de la zona de interés). Como puede observarse, conseguimos una pérdida de tasa máxima inferior al 1% manteniéndonos en un 22'5%, 40% y 65% del tiempo real absoluto, para las tareas 1952, 5000-85-15 y 10000-85-15, respectivamente.

En resumen, a la hora de evaluar la bondad del ahorro de un sistema multi-módulo frente a uno en un único paso (o de dos multi-módulos con más o menos etapas), hay que estudiar:

- El ahorro relativo en tiempo que supone el uso de una u otra estrategia, al margen de las tasas de reconocimiento alcanzables (expresión del ahorro relativo de tiempo de la página 59), lo que nos permitirá tener una idea precisa acerca de la conveniencia computacional de la estrategia de hipótesis-verificación.

1. Como se describe más en detalle en nuestra propuesta de mecanismos de evaluación y comparación, en el Apartado 3.5.1, a partir de la página 71.

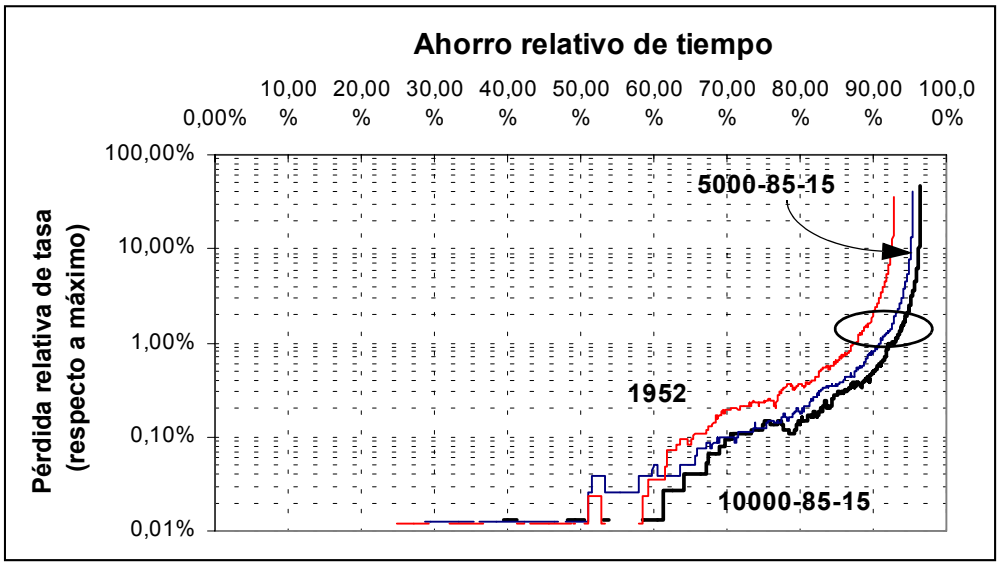

Figura 3-4: Gráfica de pérdida de tasa final de reconocimiento (para el primer candidato) en función del ahorro relativo de tiempo conseguido por un sistema en dos pasos frente a uno en un único paso

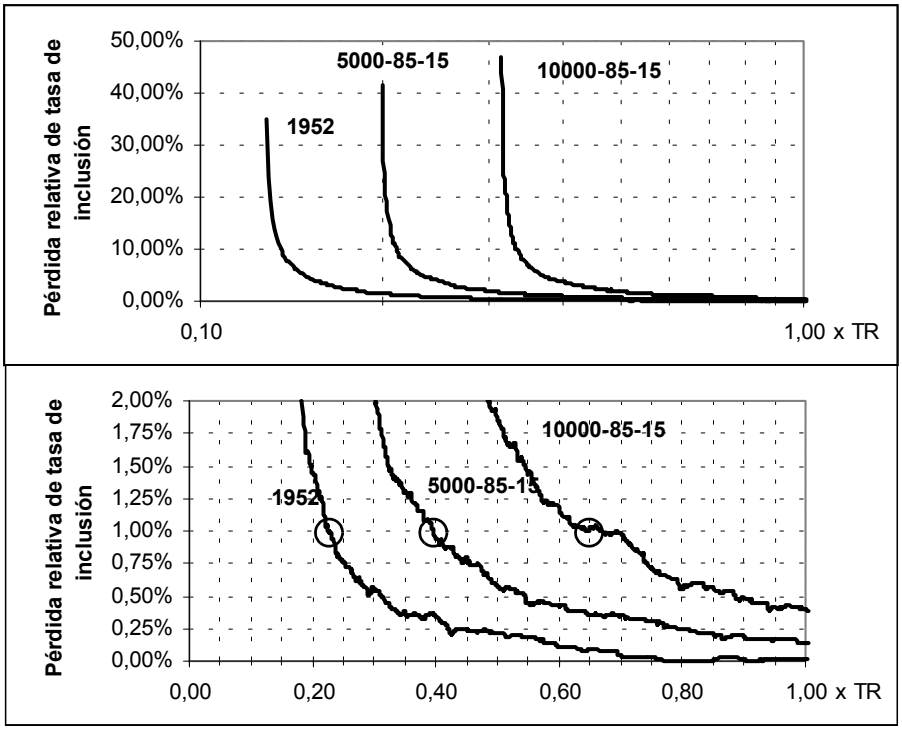

Figura 3-5: Pérdida relativa de tasa de inclusión en función de la fracción de tiempo real utilizado.

- El impacto en tasa de reconocimiento del sistema multi-módulo frente al de un sólo paso, en función del ahorro porcentual alcanzado (Figura 3-4), lo que nos dará una idea del margen de reducción alcanzable en tiempo, manteniendo la tasa por encima de un cierto valor referido al máximo absoluto conseguible por el sistema integrado.
- El impacto de la reducción de coste computacional en la tasa de reconocimiento alcanzable, medido como pérdida de tasa frente a la fracción de tiempo real alcanzable (Figura 3-5). Esta medida permite evaluar la adecuación final de una arquitectura a un

hardware concreto, mientras que las dos anteriores (fijado el hardware y las optimizaciones algorítmicas correspondientes) son independientes del mismo y serían extrapolables a nuevas situaciones de entorno computacional

3.4.3 Tasas de reconocimiento al combinar módulos

3.4.3.1 Planteamiento teórico

En este apartado estudiaremos la tasa de reconocimiento final obtenible en un sistema multi-módulo basado en el paradigma hipótesis-verificación, particularizado a reconocedores de habla aislada. Si no entramos en un análisis detallado, la primera aproximación intuitiva nos diría que la tasa final obtenible no es más que el resultado de multiplicar las tasas de cada uno de los módulos implicados:

$$\psi_{1M}(\lambda, V, V_2, V_3, \ldots, V_M) = \prod_{i=1}^{M} \psi_i(\lambda)$$

Sin embargo, *esa expresión no es válida* al no considerar el efecto de la reducción del vocabulario activo a la salida de cada módulo. De hecho, si no se impone dicha reducción (lo que, por otra parte, no tiene sentido), es decir $V_i = V, \forall i$, la tasa final será exclusivamente la obtenible por el módulo M_M[1].

Si consideramos el caso más sencillo, dos módulos concatenados, podemos plantear dos situaciones de cara a obtener la tasa de reconocimiento final $\psi_{12}(\lambda, V, V_2)$:

1. La palabra a reconocer se encuentra en la lista generada por el módulo M_1 en la posición $p_1 \le V_2$. Eso sucede con una probabilidad igual a $\psi_1(V_2)$. En este caso hay, de nuevo, dos posibilidades:
    1. El módulo M_2 acierta la palabra en la posición $p_2 \le \lambda$, lo que sucede con probabilidad igual a $\psi_2'(\lambda, V_2)$ (palabras acertadas que contribuyen a la tasa $\psi_{12}(\lambda, V_2)$)
    2. El módulo M_2 acierta la palabra en la posición $p_2 > \lambda$, lo que sucede con probabilidad igual a $1 - \psi_2'(\lambda, V_2)$ (palabras falladas).
2. La palabra a reconocer se encuentra en la lista generada por el módulo M_1 en la posición $p_1 > V_2$. Eso sucede con una probabilidad igual a $1 - \psi_1(V_2)$ y corresponde a palabras falladas finalmente

Así, la tasa final obtenida será:

$$\psi_{12}(\lambda, V, V_2) = \psi_1(V_2)\psi_2'(\lambda, V_2)$$

Como era de esperar, el efecto del módulo de preselección es vital, al depender la expresión global del valor que tome $\psi_1(V_2)$ (factor de escala sobre el comportamiento del segundo módulo). En apartados posteriores se planteará el estudio de la variación de la longitud de la lista de preselección V_2 y su efecto en la tasa conjunta.

1. Esta afirmación es válida siempre y cuando no se usen estrategias de combinación de los *scores* generados por cada uno de los módulos que intervienen, en cuyo caso podríamos llegar a obtener mejores resultados.

La generalización para M módulos es sencilla y da lugar a la siguiente expresión[1]:

$$\psi_{1M}(\lambda, V_2, V_3, \ldots, V_M) = \psi_1(V_2)\left(\prod_{i=2}^{M-1} \psi_i'(V_{i+1}, V_i)\right)\psi_M'(\lambda, V_M)$$

Que puede calcularse progresivamente a partir de asociaciones sucesivas de módulos, del 2 al M, como se describe en el Apartado 3.4.3.3 a partir de la página 67. Nuestro objetivo sería calcular el conjunto óptimo de tamaños de listas de preselección $(V_2, V_3, \ldots, V_M)$ para cumplir con un objetivo de tasa conjunta dada. Para ello necesitamos:

- En primer lugar, la curva de tasa de inclusión del primer módulo ψ_1
- En segundo, las correspondientes a los siguientes módulos, condicionadas al comportamiento del anterior (ψ_i').

El cálculo de las funciones $\psi_i(\lambda)$, y en particular la $\psi_1(\lambda)$ que usamos en la expresión general, es razonablemente sencillo ya que basta con ejecutar un experimento de reconocimiento con el primer módulo de la cadena y evaluar la tasa para distinto número de candidatos considerados, pero no sucede lo mismo con las $\psi_i'(\lambda, V_i)$, al ser, dependientes del comportamiento de los módulos M_{i-1}: no podemos permitirnos calcular todas las posibilidades de combinación de tamaños y *calidades*[2] de lista y sistemas, sino que queremos obtener información sobre si una combinación de módulos funcionará para unos objetivos dados, de forma rápida. Discutamos un poco más sobre esta dependencia.

A la vista de la definición de $\psi_i'(\lambda, V_i)$ está claro que se trata de la tasa de reconocimiento (o inclusión, al ir variando λ) del módulo M_i para un vocabulario dado, compuesto por V_i palabras, de modo que el comportamiento de $\psi_i'(\lambda, V_i)$ depende fuertemente del vocabulario dado V_i. Hasta aquí podría parecer que las sucesivas ψ_i' dependen únicamente del tamaño del vocabulario V_i. Si así fuera, bastaría con ejecutar una serie de experimentos con el módulo de verificación sobre distintos tamaños de vocabulario para obtener la curva final, sin necesidad de hacer la combinación de módulos en la práctica (llamemos a esa curva ψ_i''). Sin embargo, esto no es posible: V_i *no* es un vocabulario cualquiera dado que el módulo de hipótesis hará un trabajo razonable en el sentido de preseleccionar las palabras más parecidas a la producida. En resumen, V_i es una lista *especialmente complicada* de palabras, al ser un vocabulario de tamaño dado en el que las palabras están ordenadas con un criterio de *similaridad* a partir de las fuentes de información con las que contamos. Eso hace que no sea posible calcular ψ_i' sin pasar por la experimentación. La curva ψ_i'' sería una estimación *optimista* de ψ_i' que nos puede valer para observar la tendencia general de la combinación[3]

Sin embargo, el dato más importante acerca de ψ_i' es que podemos acotarla con relativa facilidad, sin necesidad de recurrir a una experimentación costosa: sabemos que su máximo es el 100% y que su valor mínimo es la tasa de reconocimiento alcanzable cuando enfrentamos al módulo de verificación con el vocabulario completo. Así, una cota inferior para $\psi_{12}(\lambda, V_2)$ (volvemos al caso de una arquitectura en dos pasos) será:

$$\psi_{12}(\lambda, V_2) \geq \psi_1(V_2) \cdot \min_{V_2}\{\psi_2'(\lambda, V_2)\}$$

1. En lo que sigue, no volveremos a incluir la dependencia del vocabulario total V en $\psi_{12}(\lambda, V, V_2)$, por comodidad, ya que V no aparece explícitamente en las fórmulas, aunque está implícita en el valor de $\psi_1(\lambda)$. Así, hablaremos de ahora en adelante de $\psi_{12}(\lambda, V_2)$.
2. A qué nos referimos con esta idea de *calidad* quedará más claro en los párrafos siguientes.
3. Podríamos discutir acerca de la mejor estrategia para diseñar los vocabularios a usar en la construcción de ψ_i'', desde la más simple aleatoria, hasta la que se orientaría a la selección de palabras con criterios de confusabilidad, pasando por criterios frecuenciales, etc.

siendo:

$$min\{\psi_2'(\lambda, V_2)\} = \psi_2'(\lambda, V) = \psi_2(\lambda) \geq \psi_2(1)$$

En general no estamos interesados en cualquier valor de λ: salvo que estemos estudiando el comportamiento de arquitecturas con más de dos módulos, nos interesará $\lambda = 1$, con lo que podremos calcular $\psi_{12}(1, V_2)$, es decir, la tasa de acierto del sistema conjunto para el primer candidato, en función de la longitud de lista de preselección. Así, su valor quedará:

$$\psi_{12}(1, V_2) \geq \psi_1(V_2)\psi_2(1) = \psi_{12pesimista}(1, V_2)$$

Esta última expresión nos indica que, en el caso peor, la tasa de acierto del sistema final vendrá dada por la multiplicación de dos valores calculables con relativa facilidad: las tasas de inclusión de los módulos que forman el conjunto, evaluadas en dos puntos concretos, sin necesidad de hacer una costosa evaluación de los dos módulos concatenados en distintas condiciones. A partir de ese valor, y en algunos casos, será posible determinar la viabilidad algorítmica de una determinada combinación de módulos. Si la estimación de tasa obtenida es suficiente para nuestros intereses y, además, se cumplen las restricciones de tiempo impuestas por la arquitectura hardware sobre la que se hace la implementación, estaremos seguros de su correcto funcionamiento en dicha implantación real: como mínimo, funcionará con la tasa prevista, pudiendo llegar a conseguir mejoras importantes como se muestra en el ejemplo que se describe a continuación (recordamos que hemos hecho una estimación pesimista).

Veamos un ejemplo de un experimento real particularizado a dos módulos: En la Figura 3-6 se muestran los resultados obtenidos en una tarea de dos módulos sobre un diccionario de 10000 palabras (nótese que hemos usado el eje de abscisas logarítmico, para facilitar la visualización), tomando $\lambda = 1$ (mostramos la evolución de la tasa de acierto condicionada para el primer candidato[1]) y variando el tamaño del vocabulario V_2 entregado por el módulo de hipótesis (entre 1 y 10000, obviamente). La función decreciente de la parte superior es $\psi_2'(1, V_2)$ donde, como puede verse, partimos de una tasa del 100% (sólo hay una palabra en la lista de preselección y es correcta), hasta el 76'99%, que es la tasa de acierto que obtiene el módulo de verificación cuando se le presentan todas las palabras del diccionario $(\psi_2(1))^2$. La función creciente que parte de alrededor de un 41% y llega al 100% es $\psi_1(V_2)$, la curva de tasa de inclusión del módulo de preselección. El producto de ambas es, por supuesto, $\psi_{12}(1, V_2)$. La curva adicional creciente que parte de un 76'99% y llega al 100% es la curva de tasa de inclusión considerando directamente el módulo de verificación, enfrentado al diccionario completo (obviamente el punto inicial $\psi_2(1)$ coincide en valor con el final $\psi_2'(1, V_2)$).

Si hubiéramos aplicado el cálculo aproximado para la función $\psi_2'(1, V_2)$ descrito anteriormente, habríamos llegado a la situación mostrada en la Figura 3-7 en la que aparece la curva real de tasa conjunta para el primer candidato, la aproximada y la reducción relativa en tasa de error en la que nos encontraríamos en este ejemplo si lo implementáramos, con respecto a la aproximación (pesimista).

3.4.3.2 Consideraciones sobre el número de candidatos a preseleccionar

En otros capítulos de esta tesis se hace un importante esfuerzo en tratar de alcanzar altas tasas de inclusión en los módulos de preselección, como el objetivo a cumplir para que esas etapas iniciales no perjudiquen el rendimiento final conjunto. La aportación de este capítulo a esa intención inicial es fundamental, ya que marca los límites a considerar cuando tratamos de sistemas multi-etapa: tan importante es el rendimiento del módulo de preselección como el del de verificación, de modo que un potentísimo módulo de preselección puede ser inútil si se combina con uno de verificación de calidad

1. Nos centramos en el estudio del primer candidato dado que en un sistema de dos módulos es el punto más interesante. Más adelante en este capítulo discutiremos brevemente casos con más módulos.
2. Y que es la que podremos utilizar como cota inferior para $\psi_2'(1, V_2)$

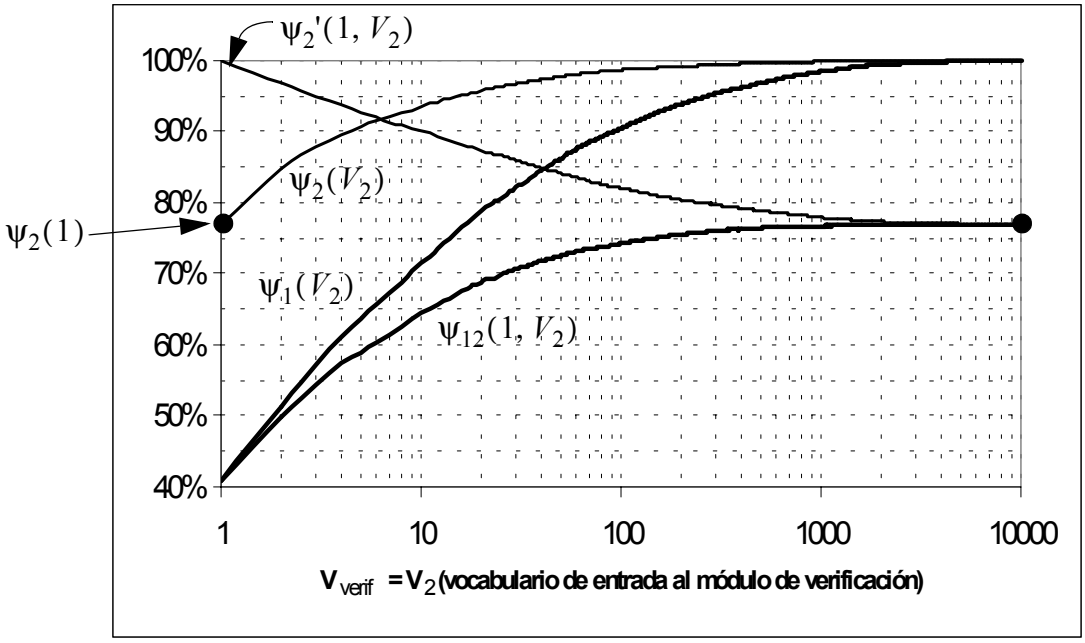

Figura 3-6: Ejemplo del comportamiento de un sistema de dos módulos (hipótesis-verificación) en un experimento real

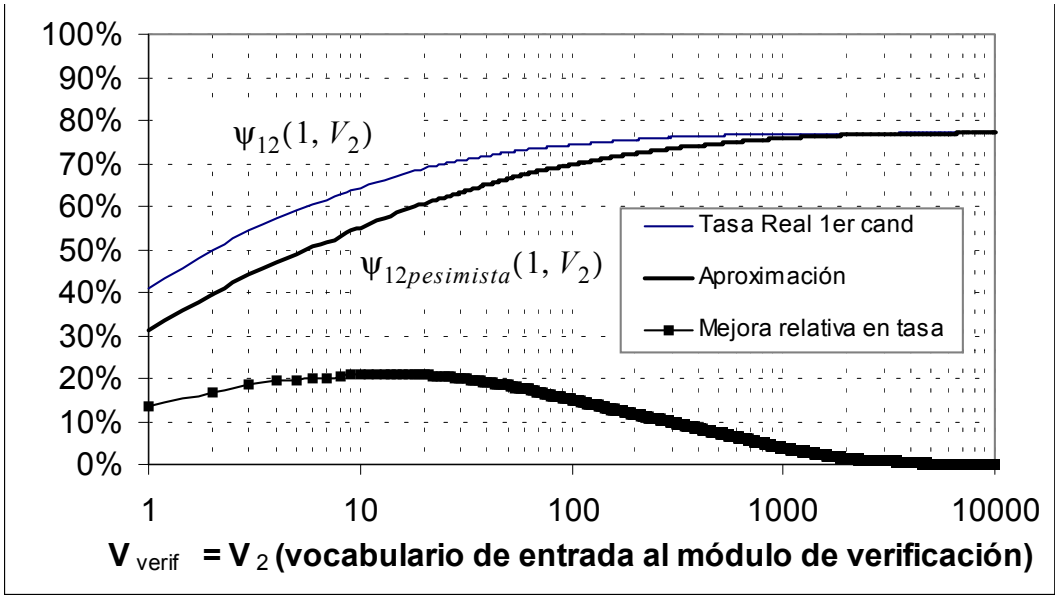

Figura 3-7: Comparación entre la curva de tasa conjunta real para el primer candidato (Tasa Real 1er cand) y la simulada (Aproximación) con el método propuesto, junto con la mejora en la tasa de error conjunta entre la situación real y la aproximación utilizada.

media (entendiendo por inútil el que finalmente no consigamos un sistema conjunto con un rendimiento adecuado a las exigencias de nuestra tarea).

En el ejemplo mostrado en la Figura 3-6, el sistema de preselección consigue una tasa del 98% en el candidato número 774. Sin embargo, la tasa de reconocimiento del conjunto hipótesis+verificación para la primera posición para dicho tamaño de lista de preselección (V_2=774) es del 76'69% ($\psi_{12}(1, 774)$), muy cerca de la máxima tasa alcanzable por el módulo de verificación en el primer candidato, enfrentado con la totalidad del diccionario ($\psi_2(1) = 76'99\%$). En este punto cabe hacer dos comentarios:

- Si el sistema final de verificación no es capaz de enfrentarse a la tarea completa (diccionario completo) con las tasas requeridas, no tiene sentido invertir esfuerzo en un buen módulo de hipótesis para diseñar un sistema en dos etapas (cuyo principal objetivo es reducir el coste computacional sin perder tasa).
- El sistema de preselección debe ser, en cualquier caso, lo mejor posible, ya que, como se verá un poco más adelante, es también determinante en el correcto rendimiento del conjunto y en no limitar la tasa final alcanzable.

Estas consideraciones parecen contradictorias: por un lado despreciamos el diseño del módulo de preselección y por otro insistimos en la importancia de conseguir altas tasas en el mismo. La lectura a hacer en este caso es conjunta: necesitamos un buen sistema de preselección y un buen sistema de verificación.

A la luz de las características del módulo de verificación que utilicemos, habrá que plantear un tamaño razonable de lista de preselección. En la Figura 3-8 se muestra la diferencia porcentual de error entre mejor tasa alcanzable por el módulo de verificación (que supone el mejor valor que podemos conseguir en el sistema conjunto) y el que de hecho se alcanza en la arquitectura hipótesis+verificación, para tres tareas dadas (diccionarios de 1952, 5000 y 10000 palabras), en función del tamaño relativo de la lista de preselección. Como puede observarse, un valor razonable de longitud máxima de lista de preselección podría estar en torno al 10%, para el que la pérdida relativa de error no supera el 3%, en el peor caso. Las consideraciones respecto a coste computacional del Apartado 3.4.2 complementan esta estimación, al fijar un límite adicional al rango de validez práctica del conjunto. Como comentario adicional sobre la Figura 3-8, resulta sorprendente cómo se obtienen mejores resultados (menor pérdida de tasa para la misma longitud de lista de preselección) a medida que pasamos a tareas de mayor tamaño de diccionario. La explicación a esta aparente paradoja radica en el uso del eje de abscisas porcentual (referido al tamaño del diccionario en cada caso) y a consideraciones adicionales sobre la dificultad de dichos diccionarios, lo que se discute en el Apartado 5.4 a partir de la página 170.

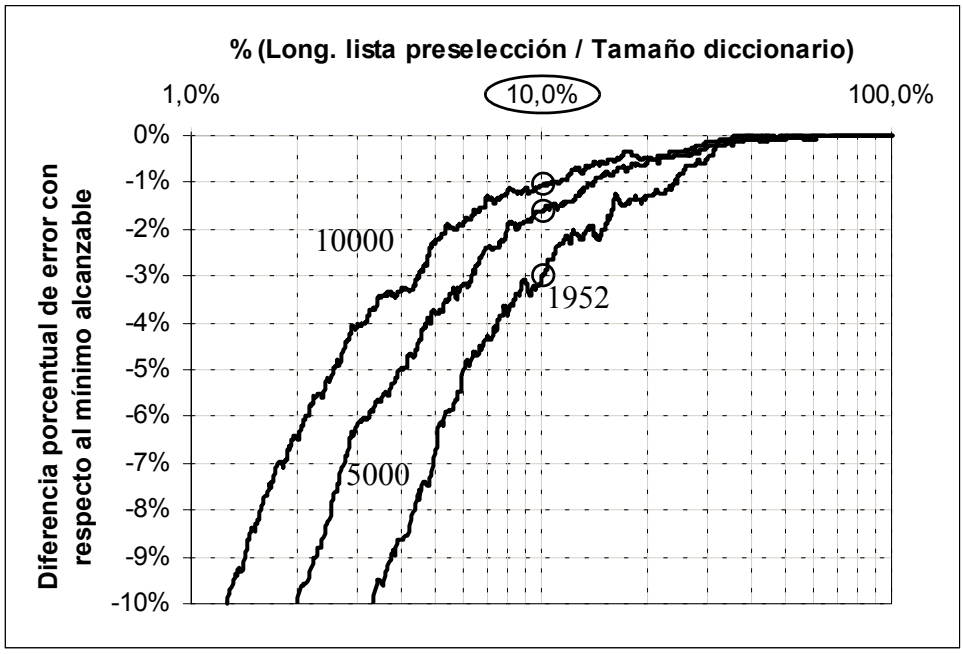

Figura 3-8: Diferencia porcentual de error entre la mejor tasa alcanzable por el módulo de verificación (límite del conjunto) y el resultado efectivo de la combinación, para tres diccionarios diferentes. El eje de abscisas muestra el tamaño de la lista de preselección normalizado por el tamaño del diccionario.

3.4.3.3 Consideraciones para más de dos módulos

Cuando el número de módulos sube por encima de dos, la formulación y la extracción de pautas se complica notablemente, al necesitar estimar dos o más tamaños de vocabulario para cada una de las etapas de preselección usadas. La decisión al respecto tendría que ser tomada introduciendo factores adicionales.

Para el caso de tres módulos, por ejemplo, tenemos:

$$\psi_{13}(\lambda, V_2, V_3) = \psi_1(V_2)\psi_2'(V_3, V_2)\psi_3'(\lambda, V_3) = \psi_{12}(V_3, V_2)\psi_3'(\lambda, V_3)$$

Donde como puede verse aparece el término $\psi_{12}(V_3, V_2)$ que es calculable a partir de lo visto anteriormente particularizado para dos módulos. En la Figura 3-9 se muestran curvas

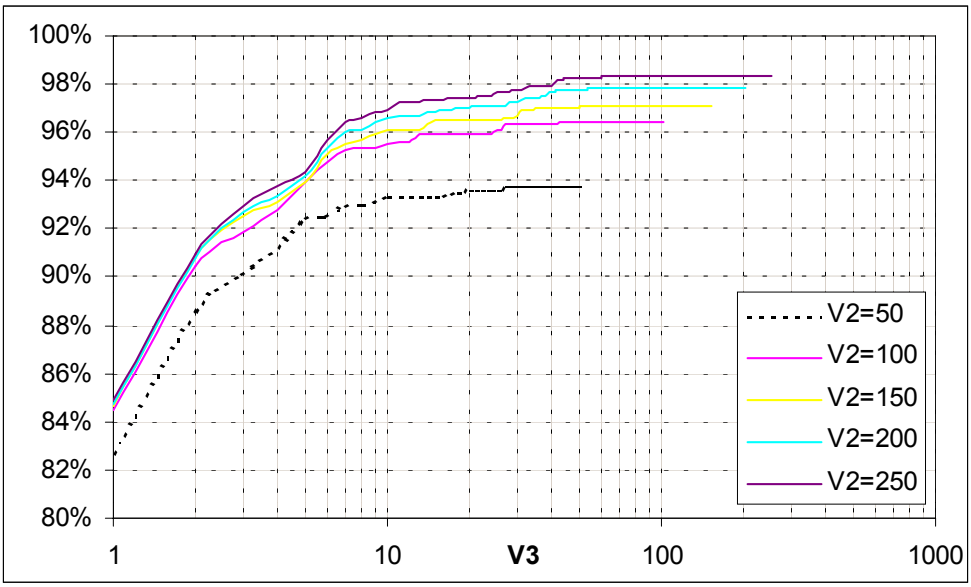

Figura 3-9: Gráficas de $\psi_{12}(V_3, V_2)$ en función de V_3 para distintos valores de V_2 (50, 100, 150, 200 y 250).

correspondientes a dicho término, para la tarea sobre 1952 palabras, en función de V_3 y para distintos tamaños de la lista de preselección (V_2) entregada al segundo módulo de la cadena (nótese de nuevo el eje de abscisas logarítmico). Obviamente, sólo se ofrecen valores para $V_3 \leq V_2$. En este punto cabe hacer las mismas consideraciones que hicimos para el caso de dos módulos (de nuevo con $\lambda = 1$ para ver el efecto en la tasa para el primer candidato del sistema final), sustituyendo la curva de tasa de inclusión visto entonces por estas nuevas curvas. Igualmente podríamos discutir acerca de la forma de la curva $\psi_3'(\lambda, V_3)$ y su relación de dependencia con los módulos previos. El diseñador tendría que optar por aquella curva de las mostradas que le garantizara unos resultados finales dentro de los márgenes de tasa decididos.

Finalmente cabe comentar que la misma estimación de cota inferior que vimos en el Apartado 3.4.3.1 puede extenderse al caso multi-módulo, quedando:

$$\psi_{1M}(1, V_2, V_3, \ldots, V_M) \geq \psi_1(V_2) \prod_{i=2}^{M} \psi_i(1)$$

Con lo cual sería relativamente sencillo calcular esa cota inferior de la tasa obtenible por el sistema multi-módulo.

3.4.4 Aplicación del enfoque teórico al diseño de un sistema basado en hipótesis-verificación no construido

A lo largo de este capítulo se han dado pautas y consideraciones de cara al diseño de sistemas multi-módulo. A modo de aplicación de todo lo visto, describiremos en este capítulo el diseño de un sistema basado en el paradigma hipótesis-verificación así como su evaluación, sin necesitar de la construcción física del mismo.

Para ello partiremos de los sistemas descritos e implementados en el Apartado 3.3.2 para el módulo de hipótesis (generación de cadena fonética seguido de acceso léxico) y en el Apartado 3.3.1 para el de verificación (búsqueda acústica basada en el algoritmo de un paso guiado con árbol), en todos los casos usando modelado semicontinuo independiente del contexto con el alfabeto `alf45`. Nuestro objetivo es decidir acerca de la adecuación de dichos módulos para hacer frente a la tarea POLYGLOT[1], en la que el diccionario usado está compuesto por 2000 palabras.

Nuestros datos de partida son aquellos fácilmente calculables o impuestos por la tarea:

- Curva de tasa de inclusión para el módulo de hipótesis enfrentado al diccionario completo $\psi_1(V_2)$ (la mostrada en la Figura 3-10, siendo el eje de abscisas V_2, medido en función de la longitud de la lista de preselección calculada como el porcentaje sobre el tamaño del diccionario).
- Curva de tasa de inclusión para el módulo de verificación enfrentado al diccionario completo $\psi_2(V_2)$ (mostrada también en la Figura 3-10). La tasa para el primer candidato, $\psi_2(1)$, es de un 93'71%.
- Tiempo de proceso medio del módulo de hipótesis: 63'3 milisegundos. por palabra para procesar todo el diccionario ($\tau_1 \cdot V$).
- Tiempo de proceso medio del módulo de verificación: 15 segundos por palabra para procesar todo el diccionario ($\tau_2 \cdot V$). Obviamente este tiempo es absolutamente inaceptable para cualquier aplicación y es el obtenido por dicho módulo teniendo en cuenta que no se ha aplicado ningún tipo de optimización algorítmica.
- Admitiremos una pérdida de tasa inferior al 1% con respecto al máximo obtenible en nuestro caso (93'71%), por ejemplo, es decir, buscamos obtener un 92'8% en el primer candidato del sistema conjunto, como mínimo.
- La duración media de las palabras de la base de datos es de 420 milisegundos, lo que marca el valor de proceso en tiempo real a conseguir.

A partir de estos datos, trabajaremos de acuerdo con lo descrito en el desarrollo teórico. En primer lugar, la tasa alcanzable para el primer candidato por el sistema conjunto será (aproximación pesimista):

$$\psi_{12}(1, V_2) \geq \psi_1(V_2)\psi_2(1) = \psi_1(V_2) \cdot 0'9371$$

que se muestra también en la Figura 3-10, en la que puede apreciarse la saturación para pequeños valores de longitud de lista de preselección (entre el 1% y el 10% del tamaño del diccionario). Los valores iniciales de la gráfica (abscisa 0'05%) corresponden a la tasa obtenida para el primer candidato (1/2000 = 0'0005). En la Figura 3-11 se muestra la pérdida relativa de tasa conseguida, con respecto al máximo obtenible ($\psi_2(1)$ = 93'71%), también en función de la longitud de la lista de preselección calculada como el porcentaje sobre el tamaño del diccionario.

Hasta aquí, las consideraciones respecto a pérdida de tasa nos permitirían asegurar un funcionamiento del sistema conjunto muy cercano al de un único paso (y recordando siempre que utilizamos una aproximación pesimista):

1. Base de datos limpia compuesta por 30000 producciones de habla aislada correspondientes a 30 locutores, descrita en detalle en el Anexo B.3 a partir de la página 192. El diccionario de la tarea propuesta está compuesto por 2000 palabras.

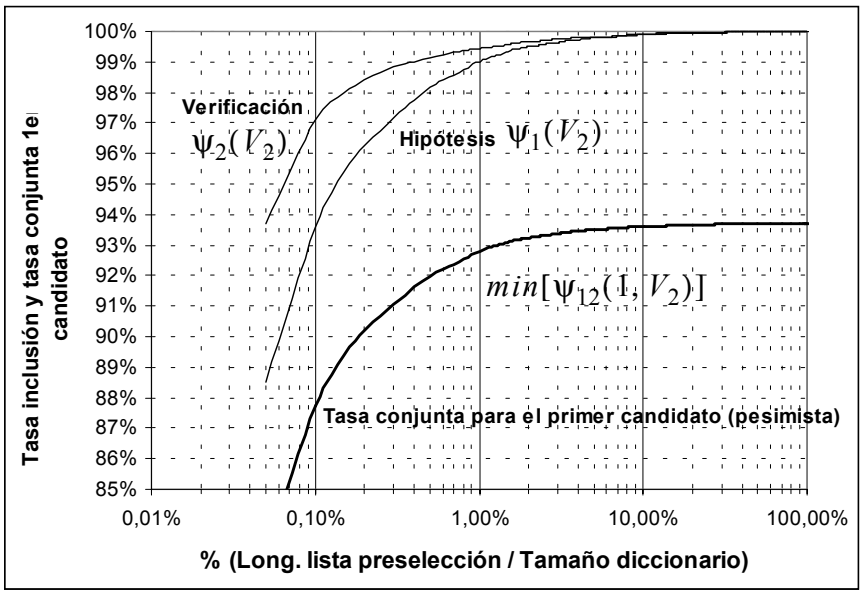

Figura 3-10: Tasa de error de inclusión para el módulo de hipótesis (un paso y acceso léxico) y el de verificación (integrado) sobre la tarea POLYGLOT. Tasa conjunta para el primer candidato. Modelado semicontinuo independiente y dependiente del contexto con `alf23`.

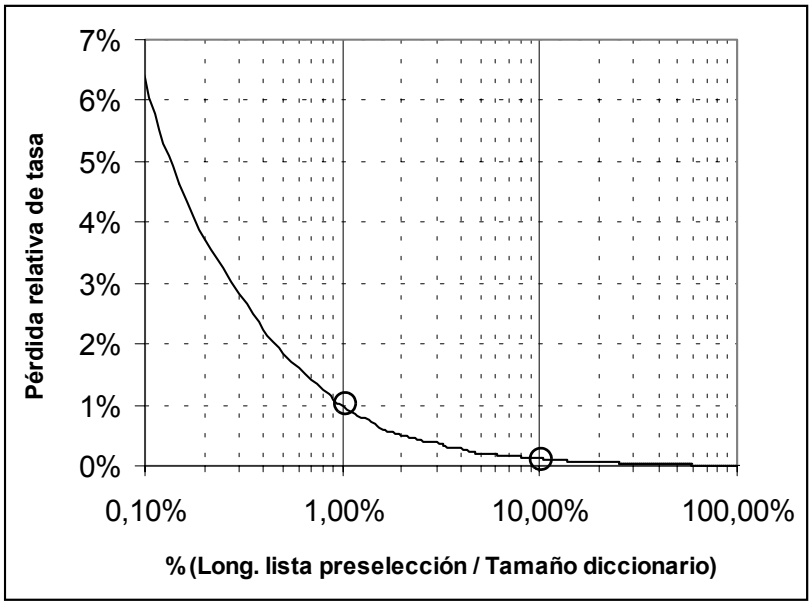

Figura 3-11: Pérdida relativa de tasa para el primer candidato en el sistema conjunto, en función de la longitud de lista de preselección.

- Para el objetivo marcado anteriormente: Con una pérdida de tasa inferior al 1% (que supone sin embargo un incremento de error de un 14'75%), usando una longitud de lista de preselección de 20 candidatos (1% del tamaño del diccionario)
- Para un nuevo objetivo (para mostrar otro punto relevante): Con una pérdida de tasa inferior al 0'1% (incremento de error de un 1'49%) usando una longitud de lista de preselección de 200 candidatos (10% del tamaño del diccionario)

Aún nos falta introducir la dimensión de coste computacional. Con los valores de tiempo de proceso vistos más arriba, el módulo de verificación tomado como único sistema de reconocimiento trabaja a 36 veces tiempo real sobre el hardware en el que se ejecutaron los experimentos (más de 15 segundos de tiempo de proceso por palabra). El módulo de hipótesis es unas 238 veces más rápido que éste último. En la Figura 3-12 se muestra la disminución relativa de tasa para el primer candidato en

función del ahorro de tiempo conseguible por el sistema conjunto[1], en comparación con el uso de un único paso (similar a la Figura 3-4). Finalmente, en la Figura 3-13 (similar a la Figura 3-5) se muestra la pérdida relativa de tasa alcanzable en función de la fracción de tiempo real utilizada por el sistema.

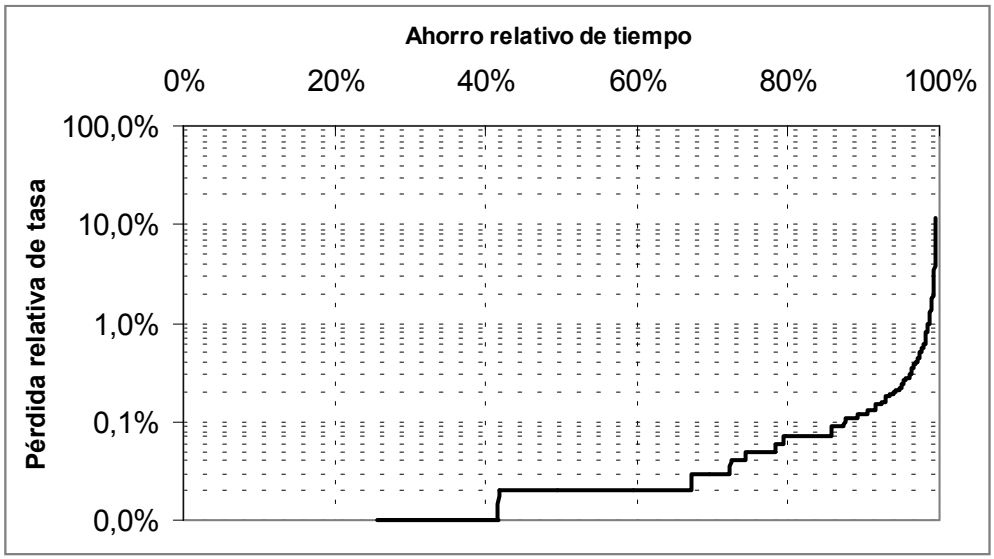

Figura 3-12: Disminución relativa de tasa para el primer candidato en función del ahorro de tiempo conseguible por el sistema.

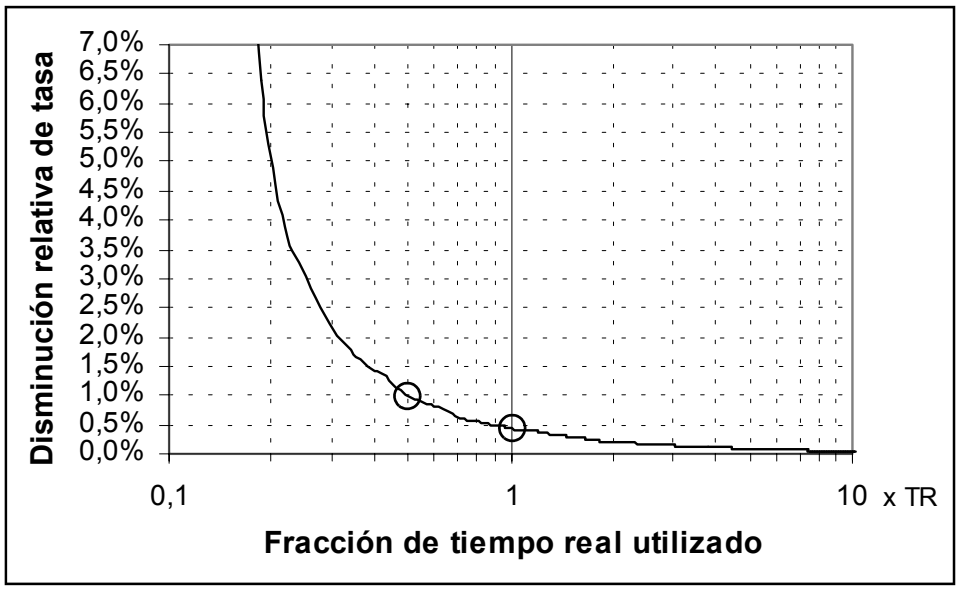

Figura 3-13: Disminución relativa de tasa para el primer candidato en función de la fracción de tiempo real usada en el sistema.

Como puede observarse, podemos trabajar en tiempo real asumiendo en este caso una pérdida relativa de tasa de un 0'5%, lo que haría a nuestro sistema conjunto cumplir todos los requisitos necesarios para su aplicabilidad a nuestra tarea, sobre todo teniendo en cuenta que el sistema real (una vez implementado) tendrá unas prestaciones superiores a las mostradas.

Por último, haremos una consideración sobre el uso diagnóstico adicional de nuestro desarrollo. Gráficas como la de la Figura 3-12 valen al diseñador de sistemas de reconocimiento como herramienta de especificación de requisitos: Así por ejemplo, si nuestro requisito de partida hubiera

1. La gráfica aparece escalonada para valores bajos de ahorro y pérdida de tasa ya que esos puntos corresponden a valores elevados de tasa de inclusión, en la que la variación de un sólo ejemplo acertado implica una gran subida en la longitud de lista necesaria.

sido obtener un incremento relativo máximo de *tasa de error* del 5%, habríamos recurrido a la Figura 3-14 en la que se muestra el incremento de dicha tasa de error en función de la fracción de tiempo real utilizada. A partir de ella es evidente que no cumplimos los requisitos de tiempo real, de modo que el diseñador tendría que trabajar en la optimización del módulo de verificación para reducir su tiempo medio de proceso y así llegar a cumplir su objetivo. A partir de la formulación teórica desarrollada es fácil estimar el valor exacto de tiempo que necesitaríamos para cumplir las nuevas restricciones. Teniendo en cuenta los valores particulares del ejemplo mostrado y modificando únicamente el tiempo medio de proceso del módulo de verificación, se ha calculado que bastaría reducir dicho tiempo de 15 a unos 11 segundos por palabra para llegar al objetivo impuesto de tasa de error en tiempo real. El uso de las gráficas y el planteamiento teórico ayudan en este caso a obtener requisitos precisos que asegurarán la adecuación del sistema a la tarea.

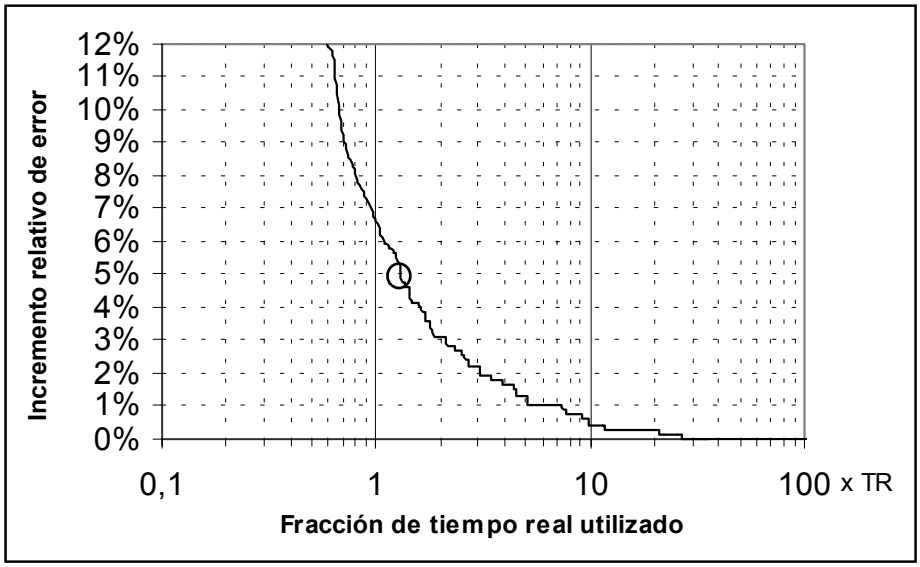

Figura 3-14: Incremento relativo de tasa de error para el primer candidato en función de la fracción de tiempo real usada en el sistema.

3.5 Experimentación sobre arquitecturas

En este apartado abordaremos el estudio del rendimiento de cada una de las arquitecturas diseñadas e implementadas sobre dos tareas radicalmente distintas: VESTEL-L, independiente del locutor sobre línea telefónica[1]; y POLYGLOT[2], dependiente del locutor en un entorno de habla limpia. Como se introdujo en su momento, la idea es estudiar las prestaciones obtenidas y la adecuación de las arquitecturas diseñadas a esas tareas.

El modelado y el alfabeto utilizado será especificado en cada caso, siendo el seleccionado como más adecuado el hallado a partir de la experimentación descrita en el Capítulo 5, a partir de la página 141.

3.5.1 Propuesta de mecanismo de evaluación

Dado que nos encontramos en muchos de los sistemas sobre los que se ha trabajado con módulos de preselección, es razonable mostrar curvas de tasa de inclusión (o de error de inclusión), esto es, representaciones de la tasa alcanzable en función del número de candidatos seleccionados.

1. Que se describe en detalle en el Anexo B.2 a partir de la página 189.
2. Que se describe en detalle en el Anexo B.3 a partir de la página 192.

Sin embargo, y con el objetivo fundamental de facilitar la comparación entre tareas que usan diccionarios con distinto número de entradas y reforzar la visión de *preselección*, proponemos modificar el eje de abscisas de las curvas mencionadas, normalizando dicha longitud de lista por el tamaño del diccionario usado.

Éste será el enfoque usado en el resto de este documento y sobre dicho eje se mostrarán los valores pertinentes, ya sean tasas de acierto, de error, mejoras relativas, etc.

Proponemos igualmente otra forma alternativa de evaluación: cuando se trata de hacer comparaciones del impacto de ciertas modificaciones en distintos sistemas (sobre todo con variaciones arquitecturales), la medida típicamente usada es la variación relativa de error. El problema es evaluar dichos impactos sobre tasas muy distintas del sistema base. La argumentación tradicional establece que precisamente para ello se usa esa medida de variación relativa y que basta con dar la tasa del sistema base para tener la perspectiva completa. Nuestra propuesta al respecto es ofrecer medidas de mejora relativa *en función de la tasa de error del sistema base*, con lo que en una misma gráfica tenemos toda la información necesaria. Esto es especialmente importante en comparaciones para distintas longitudes de lista de preselección en sistemas basados en el paradigma hipótesis-verificación: las comparaciones directas de mejora relativa en tasa de error en función del tamaño de dicha lista (medida en valor absoluto o como porcentaje sobre el tamaño del diccionario usado) son difíciles de analizar, mientras que la medida propuesta permite una visión alternativa muy interesante, en nuestra opinión.

La justificación de su uso es la siguiente: cuando comparamos sistemas similares con puntos de trabajo (tasas) similares, es razonable basar dicha comparación en la mejora relativa en la tasa de error del sistema, por ejemplo. Sin embargo, si dichos puntos de trabajo no son tan similares o si, directamente, están alejados, una representación más fiel sería aquella que relacionara el punto de trabajo (la tasa de error) con la mejora obtenida. Analíticamente podríamos verlo con un ejemplo: Dos sistemas diferentes tienen tasas de error del 40% y del 20% en una tarea y del 10% y del 5% en otra, respectivamente. La medida tradicional de mejora relativa de error (siendo e_i la tasa de error del sistema *i)*:,

$$\%\text{MejoraRelatErr} = 100 \cdot \frac{e_1 - e_2}{e_1}$$

nos diría que en ambos casos hemos reducido el error en un 50%, pero la mejora más complicada es la producida por el segundo, al contar con menores tasas de error de partida. Si referenciáramos dicha mejora del 50% a las tasas de partida según la expresión:

$$\frac{\%\text{MejoraRelativaError}}{e_1} = 100 \cdot \frac{e_1 - e_2}{e_1^2}$$

obtendríamos un valor de 2’5 para el primer caso y de 5 en el segundo, lo que nos podría valer como medida de referencia para indicar el mejor comportamiento relativo de éste último. Sin embargo, en lugar de usar explícitamente ese factor ($\frac{\%\text{MejoraRelativaError}}{e_1}$), proponemos el uso de gráficas como la de la Figura 3-22, que describiremos en detalle en su momento, en la que extendemos la evaluación de la mejora relativa de error en función de los distintos valores de tasa de error conseguidos, los asociados a distintos puntos de la curva de error de inclusión.

3.5.2 Resultados

En este apartado ofreceremos los resultados de cada uno de los sistemas implementados para las dos tareas bajo estudio y distintos diccionarios. Las curvas ofrecidas en la mayor parte de los casos serán las de *tasa de error de inclusión en función de la longitud de la lista de palabras reconocida, medida como porcentaje sobre el tamaño total del vocabulario con el que nos enfrentamos*, para facilitar la comparación entre tareas de distinto vocabulario.

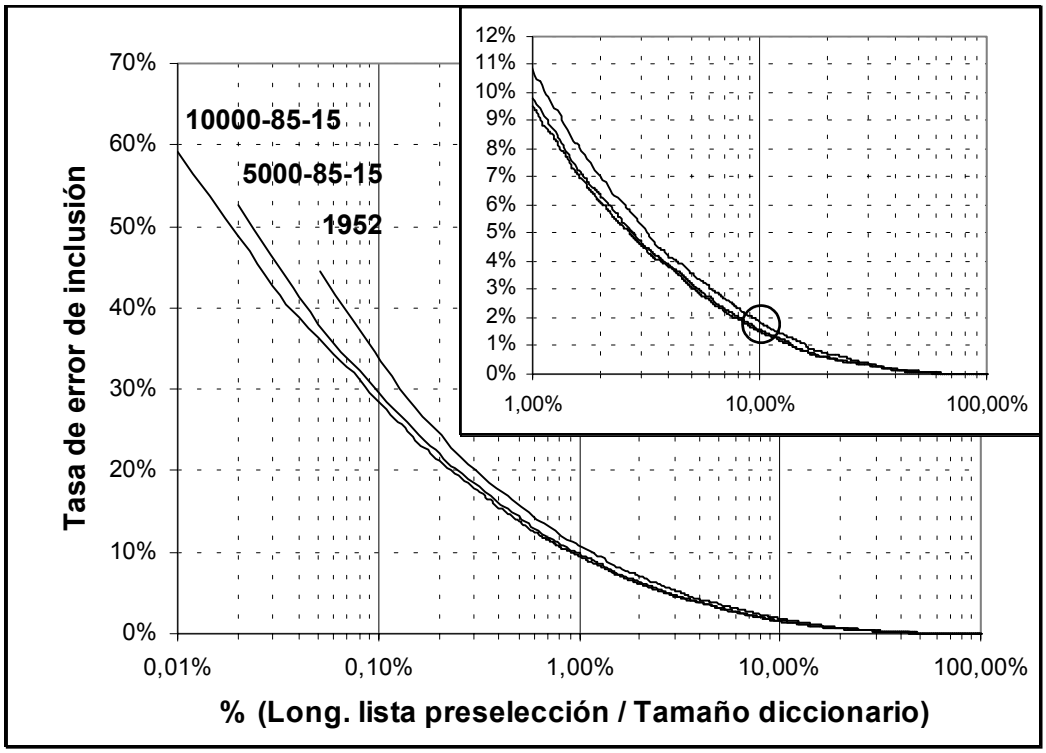

Figura 3-15: Tasa de error para el sistema no integrado (one pass y acceso léxico) sobre VESTEL-L para los diccionarios VESTEL-L1952, VESTEL-L5000-85-15 Y VESTEL-L10000-85-15. Modelado semicontinuo independiente del contexto con `alf45`.

3.5.2.1 Resultados para el sistema no integrado: Generador de cadena fonética + Acceso Léxico

En la Figura 3-15 se muestra la tasa de error obtenida con el sistema no integrado (algoritmo de un paso fonético seguido de acceso léxico) sobre la tarea VESTEL-L con los tres diccionarios definidos para la misma, el alfabeto `alf45`[1] y modelado semicontinuo independiente del contexto. La gráfica de la izquierda muestra el rango completo de valores y la de la derecha una ampliación de detalle para ver la zona de interés. Lo más destacable es la posibilidad de llegar a tasas de error de inclusión inferiores al 2% para un tamaño de lista de preselección inferior al 10% del tamaño total del vocabulario utilizado, lo que supone 195, 500 y 1000 palabras para cada diccionario, respectivamente.

En experimentos en los que el sistema no integrado evaluado en este apartado se usaba como módulo de hipótesis de una arquitectura de hipótesis-verificación (en la que el módulo de verificación era el sistema integrado que usa modelos semicontinuos dependientes del contexto con el alfabeto `alf45`), la pérdida relativa de tasa conjunta que conseguiría el sistema de hipótesis-verificación respecto al máximo alcanzable por el sistema de verificación (integrado) era de un 0'5%, 0'38% y 0'32%, respectivamente; consiguiéndose ahorros de tiempo de proceso de alrededor del 85% del tiempo que empleaba el sistema integrado enfrentado a todo el diccionario. Sin embargo, las tasas de error del sistema no integrado para el primer candidato están muy lejos todavía de poder ser considerados para la implementación de un sistema en un único paso.

En la Figura 3-16 se muestran los resultados del mismo sistema no integrado para la tarea POLYGLOT, diccionario de 2000 palabras y modelado semicontinuo independiente del contexto con el alfabeto `alf23`[2]. La tarea es comparable en tamaño de vocabulario a la de la Figura 3-15 con el diccionario de 1952 palabras pero, obviamente, se consiguen resultados mucho mejores al tratarse de

1. Compuesto por 45 unidades y descrito en detalle en el Anexo D.2.2 a partir de la página 204.
2. Compuesto por 23 unidades y descrito en el Anexo D.2.3 a partir de la página 208.

una base de datos dependiente del locutor, a pesar incluso de lo reducido de la base de datos de entrenamiento (500 palabras). En la Figura 3-16 puede observarse cómo se obtiene una tasa de error de inclusión del 1% para un tamaño de lista del 1% del tamaño del vocabulario, es decir, 20 palabras. Para obtener tasas por debajo del 2%, basta usar una longitud de lista superior al 0'5% del tamaño del vocabulario, es decir, 10 palabras. La tasa de error para el primer candidato es inferior al 12%.

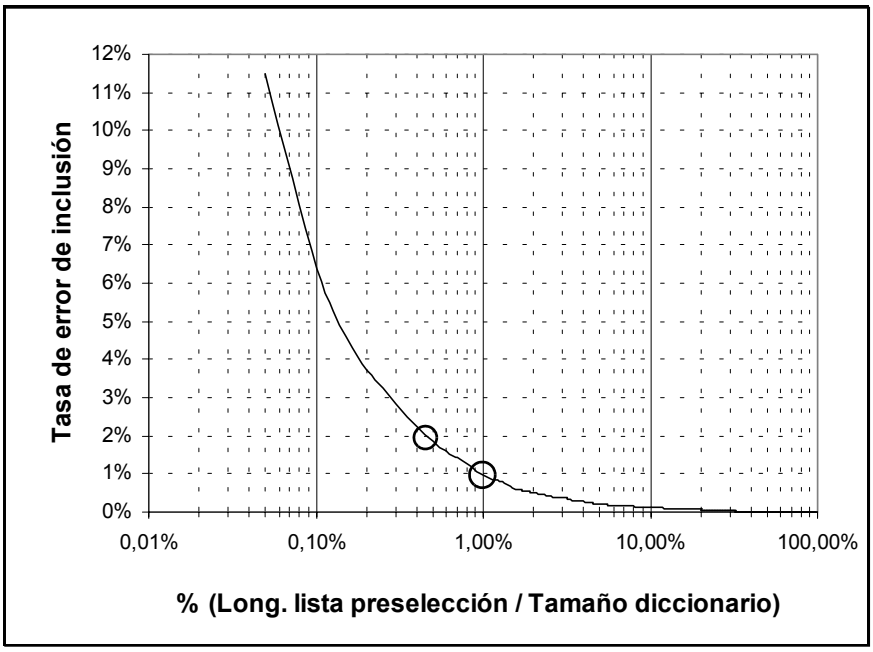

Figura 3-16: Tasa de error para el sistema no integrado (one pass y acceso léxico) sobre POLYGLOT (diccionario 2000 palabras). Modelado semicontinuo independiente del contexto con `alf23`.

De las gráficas vistas, resulta evidente que el sistema no integrado en sí, con una potencia de modelado acústico similar (modelos semicontinuos independientes del contexto) es claramente insuficiente para hacer frente a una tarea de habla telefónica independiente del locutor. Sin embargo, en el caso de la tarea de habla limpia dependiente del locutor, el sistema se acerca al que podría ser usado como sistema de reconocimiento único, sin necesidad de módulos adicionales, aunque habría que incrementar la potencia del modelo acústico usado.

3.5.2.2 Resultados para el sistema integrado con modelos independientes del contexto

En la Figura 3-17 se muestra la tasa de error obtenida con el sistema integrado (algoritmo de un paso guiado sobre un árbol) sobre la tarea VESTEL-L con los tres diccionarios definidos para la misma, el alfabeto `alf45` y modelado semicontinuo independiente del contexto. La gráfica de la izquierda muestra el rango completo de valores y la de la derecha una ampliación de detalle para ver la zona de interés. Lo más destacable es la posibilidad de llegar a tasas de error de inclusión inferiores al 2% para un tamaño de lista de preselección inferior al 2% del tamaño total del vocabulario utilizado, lo que supone 39, 100 y 200 palabras para cada diccionario, respectivamente. La tasa de error para el primer candidato es de un 27%, 34% y 41%, claramente insuficientes todavía para poder ser usados como módulo de reconocimiento único.

En la Figura 3-18 se muestran los resultados del mismo sistema integrado para la tarea POLYGLOT, diccionario de 2000 palabras y modelado semicontinuo independiente del contexto con el alfabeto `alf23`. Puede observarse cómo se obtiene una tasa de error de inclusión del 1% para un tamaño de lista del 0'4% del tamaño del vocabulario, es decir, 8 palabras. Para obtener tasas por debajo del 2%, basta usar una longitud de lista inferior al 0'2% del tamaño del vocabulario, es decir, 4 palabras. La tasa de error para el primer candidato es ligeramente superior al 6%.

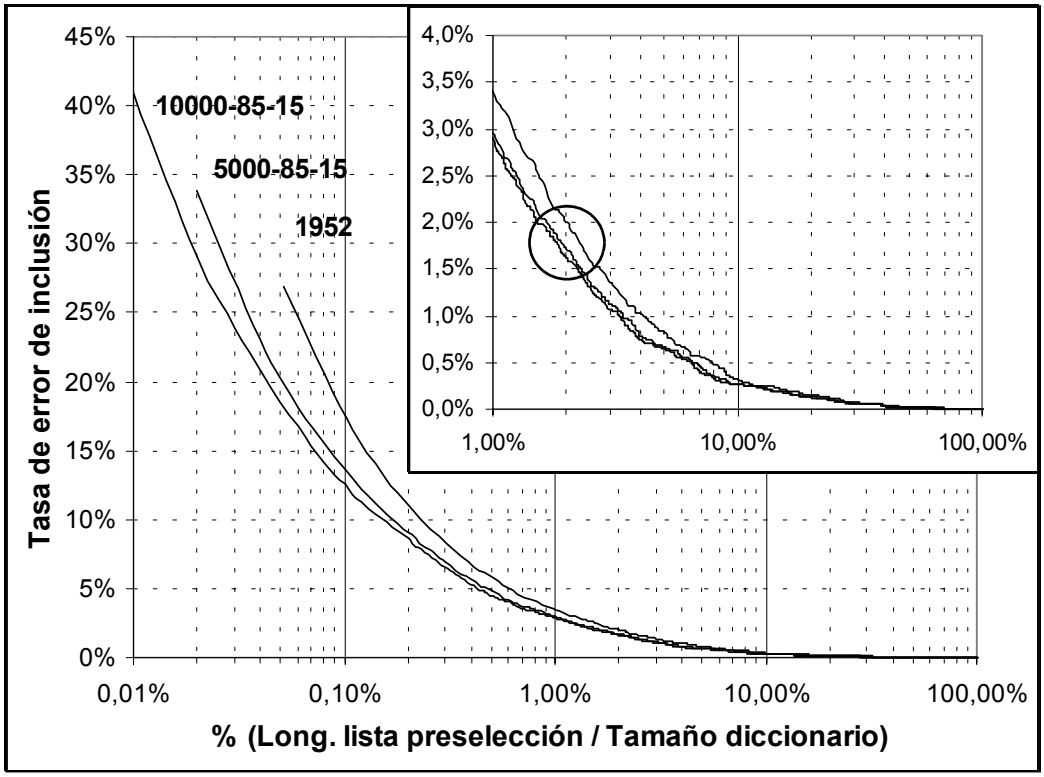

Figura 3-17: Tasa de error para el sistema integrado sobre VESTEL-L para los diccionarios VESTEL-L1952, VESTEL-L5000-85-15 Y VESTEL-L10000-85-15. Modelado semicontinuo independiente del contexto con `alf45`.

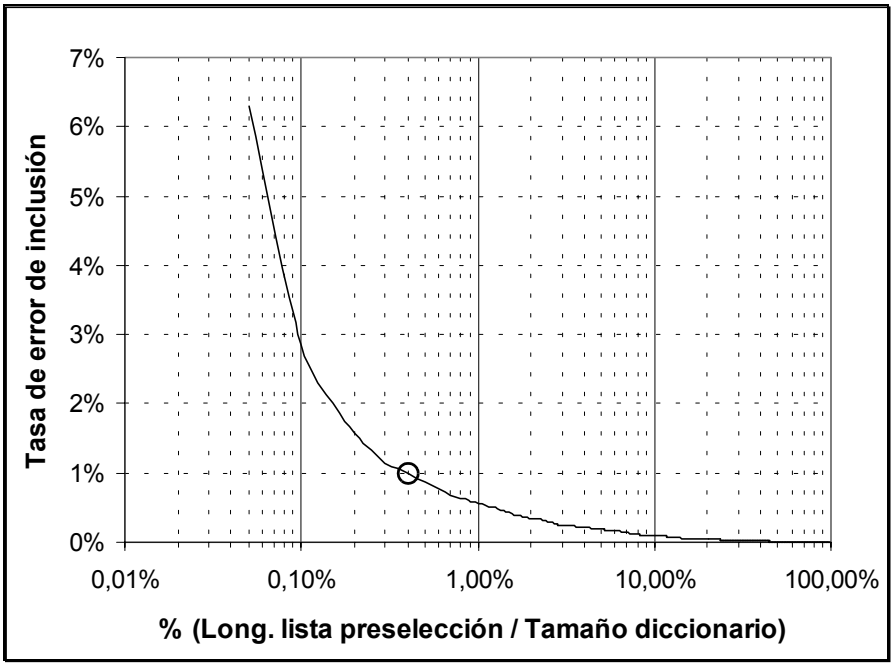

Figura 3-18: Tasa de error para el sistema integrado sobre POLYGLOT para el diccionario de 2000 palabras. Modelado semicontinuo independiente del contexto con `alf23`.

Como puede verse, a diferencia del enfoque no integrado, el mecanismo de guiado explícito durante la búsqueda acústica lleva a la obtención de resultados mucho mejores. La tarea telefónica sigue requiriendo una potencia considerablemente mayor en el modelado acústico, pero la de habla limpia podría utilizar el sistema integrado como reconocedor único, o usarlo como segundo paso de una estrategia de hipótesis-verificación.

3.5.2.3 Resultados para el sistema integrado con modelos dependientes del contexto

Dados los resultados todavía insuficientes obtenidos para la tarea de habla telefónica, nuestros esfuerzos se orientaron a la aplicación de un modelado acústico más potente: el dependiente del contexto (en el apartado Apartado 5.2.5, a partir de la página 147, se dan los detalles al respecto). Dicho modelado se aplicó sobre un sistema de búsqueda basado en el algoritmo de viterbi, con el diccionario organizado en forma lineal.

En la Figura 3-19 se muestra la tasa de error obtenida con el sistema integrado sobre la tarea VESTEL-L con los tres diccionarios definidos para la misma, el alfabeto `alf45` y modelado semicontinuo dependiente del contexto (800 distribuciones). La gráfica de la izquierda muestra el rango completo de valores y la de la derecha una ampliación de detalle para ver la zona de interés. En este caso, para conseguir tasas de error de inclusión inferiores al 2% para un tamaño de lista de preselección inferior al 0'6% del tamaño total del vocabulario utilizado, lo que supone 12, 30 y 60 palabras para cada diccionario, respectivamente (obviamente el coste computacional de este sistema es prohibitivo al tratarse de tiempos del orden de 1, 2'6 y 5'4 segundos por palabra, respectivamente[1]). La tasa de error para el primer candidato es de un 14%, 18% y 23%, todavía insuficientes para poder ser usados como módulo de reconocimiento único.

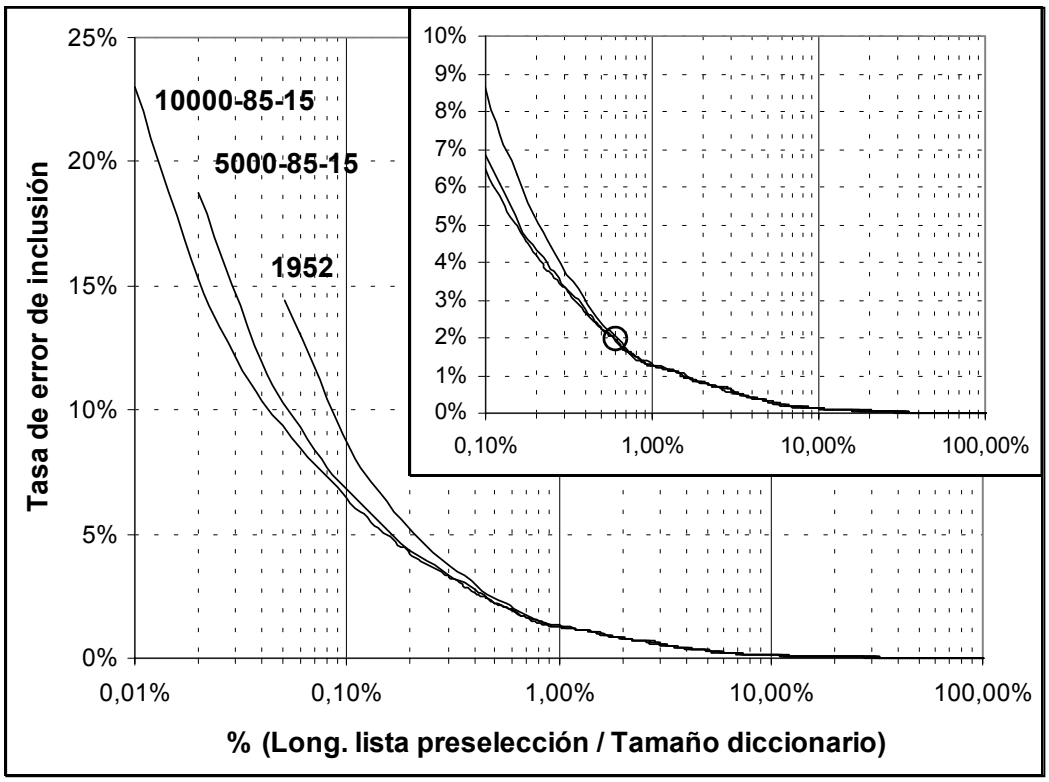

Figura 3-19: Tasa de error para el sistema integrado sobre VESTEL-L para los diccionarios VESTEL-L1952, VESTEL-L5000-85-15 Y VESTEL-L10000-85-15. Modelado semicontinuo dependiente del contexto con `alf45`.

A pesar de que no conseguimos resultados como los que serían necesarios en un sistema real, decidimos no continuar en la línea de mejorar el modelado, al no ser objeto de esta tesis, y centrarnos en la evaluación arquitectural de los enfoques vistos para cada tarea.

1. A la vista de estos tiempos, que son menores que los 15 segundos de los que hablábamos en el módulo de verificación basado en el algoritmo de un paso guiado con árbol y modelado semicontinuo independiente del contexto del Apartado 3.4.4 en la página 68, surge la duda de cómo es posible que sean efectivamente tiempos menores (ahora trabajamos con modelado semicontinuo dependiente del contexto). La explicación es que no se invirtió ningún esfuerzo en optimizar la implementación de aquel sistema y sí el aquí evaluado.

La comparación del uso de modelos dependientes e independientes del contexto se deja para el Apartado 5.2.5 a partir de la página 147.

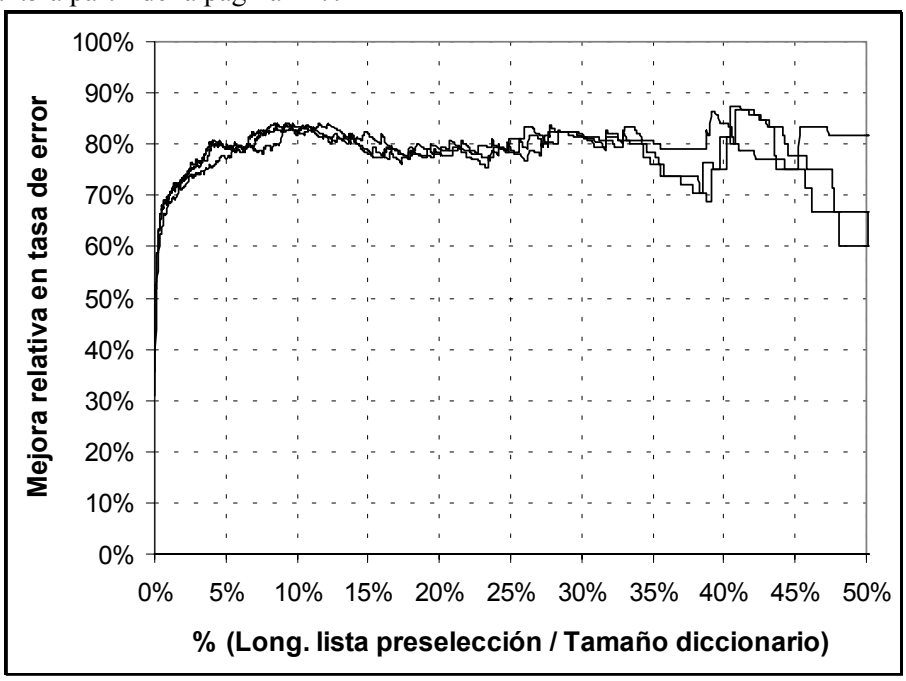

Figura 3-20: Mejora relativa en tasa de error de inclusión entre el sistema no integrado y el integrado sobre VESTEL-L para los diccionarios VESTEL-L1952, VESTEL-L5000-85-15 Y VESTEL-L10000-85-15. Modelado semicontinuo independiente del contexto con `alf45`.

3.5.3 Comparativa de arquitecturas: comparación entre el sistema no integrado y el integrado basado en modelos independientes del contexto

En este apartado se detallarán las comparativas en cuanto a cambios relativos en la tasa de error obtenida para cada sistema, con vistas a la extracción de conclusiones sobre su aplicabilidad o ventaja competitiva en distintos entornos de aplicación (tareas). Nuestro objetivo es mostrar cómo el uso de sistemas integrados en la búsqueda acústica es mucho más trascendente en tareas complejas (como la telefónica) que en tareas sencillas (como la de habla limpia). Seguiremos utilizando un eje de abscisas normalizado (longitud de lista como porcentaje sobre el tamaño del diccionario) para facilitar las comparaciones.

En la Figura 3-20 se muestra la diferencia relativa en tasa de error de inclusión entre el sistema no integrado y el integrado sobre la tarea VESTEL-L con los tres diccionarios definidos para la misma, el alfabeto `alf45` y modelado semicontinuo independiente del contexto. Como puede observarse, la mejora es muy importante (superior al 70% en un rango importante de longitudes de lista), lo que muestra la importancia de ofrecer un guiado explícito en la búsqueda acústica en una tarea tan compleja como la propuesta VESTEL-L.

En la Figura 3-21 se muestra la diferencia relativa en tasa de error de inclusión entre el sistema no integrado y el integrado sobre la tarea POLYGLOT con el diccionario de 2000 palabras, el alfabeto `alf23` y modelado semicontinuo independiente del contexto. Es fácil ver cómo la mejora al introducir el guiado explícito en la búsqueda acústica (sistema integrado) es mucho mayor en el caso de la tarea telefónica, al ser significativamente más compleja.

Sin embargo, podríamos pensar que la mejora aparente no es tal, sino que se debe a los distintos *puntos de trabajo* de cada sistema en cada tarea, es decir, que habría que eliminar la dependencia de la tasa de error obtenida. Así, en la Figura 3-22 se muestra la curva de mejora relativa de error de inclusión en función del error de tasa de inclusión (medida que proponíamos en el Apartado

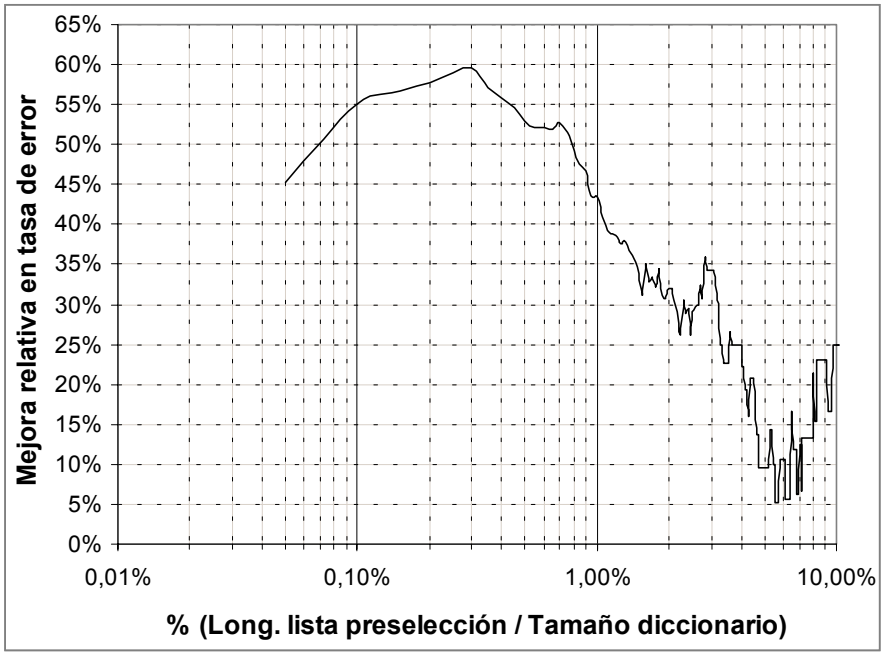

Figura 3-21: Mejora relativa en tasa de error de inclusión entre el sistema no integrado y el integrado sobre POLYGLOT para el diccionario de 2000 palabras. Modelado semicontinuo independiente del contexto con alf23.

3.5.1 de la página 71), para las tareas VESTEL-L y POLYGLOT y diccionarios de 1952 y 2000 palabras, respectivamente. Como puede verse, la mejora en la tarea telefónica sigue siendo significativamente mayor que en la de habla limpia.

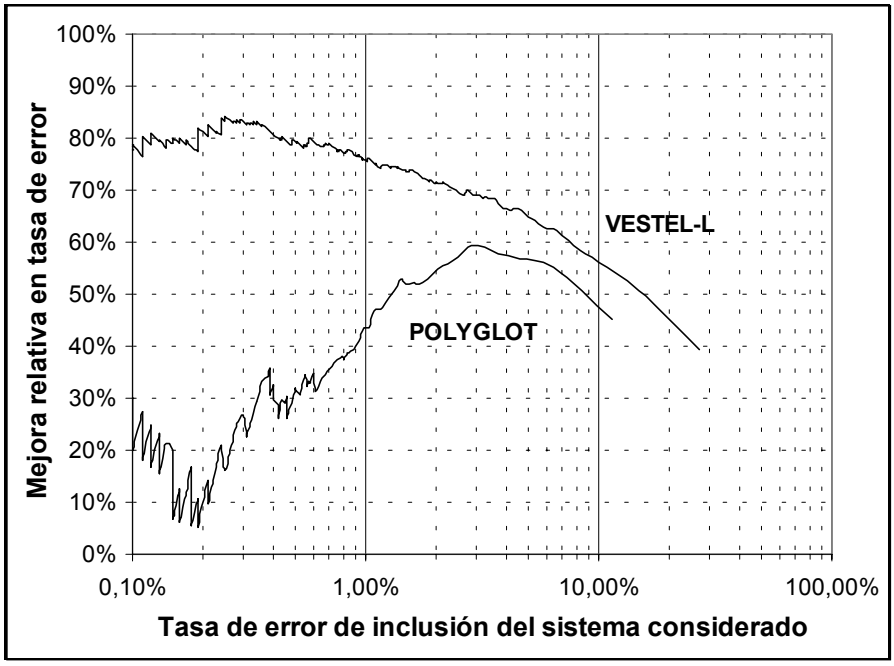

Figura 3-22: Curva de mejora relativa en tasa de error de inclusión vs. la tasa de error de inclusión entre el sistema no integrado y el integrado sobre POLYGLOT (diccionario de 2000 palabras) y VESTEL-L (diccionario de 1952 palabras). Modelado semicontinuo independiente del contexto.

Resultados similares se han obtenido en el resto de experimentos llevados a cabo en distintas condiciones de algorítmica, tamaño y tipo de vocabulario, alfabeto y tipo de modelado acústico, con lo que validamos nuestra suposición inicial acerca de la trascendencia del guiado explícito en las tareas acústicamente complejas, mucho más acusada que en las de habla limpia.

3.5.4 Comportamiento de sistemas basados en la estrategia hipótesis-verificación sobre VESTEL-L

Tras las consideraciones sobre sistemas multi-módulo del Apartado 3.4, en éste haremos referencia a los resultados obtenidos al combinar algunos de los sistemas desarrollados en una estrategia basada en el paradigma hipótesis-verificación.

Como se vio en apartados anteriores, la tarea VESTEL-L es la más compleja en cuanto a condiciones acústicas y es la que requiere mayor potencia de modelado y, por consiguiente, demanda más recursos computacionales.

Así, se decidió evaluar el comportamiento de un sistema en dos pasos formado por uno primero de hipótesis (arquitectura no integrada: one pass y acceso léxico), compuesto por el módulo cuyo comportamiento se describe en el Apartado 3.5.2.1, usando modelado semicontinuo independiente del contexto y el alfabeto `alf45`; y uno de verificación (arquitectura integrada) cuyo comportamiento se describe en el Apartado 3.5.2.3, que usa modelado semicontinuo dependiente del contexto (800 distribuciones), también con el alfabeto `alf45`.

En este caso, estamos interesados en la evaluación de la tasa de reconocimiento del sistema conjunto para el primer candidato, en función de la longitud de la lista de preselección ofrecida por el módulo de hipótesis. En las curvas inferiores de la Figura 3-23 se muestra dicho valor para los tres diccionarios considerados. En ellas puede comprobarse cómo la tasa máxima alcanzable es la del módulo de verificación enfrentado al diccionario completo (que corresponde al primer valor de las curvas superiores que son las tasas de acierto para el sistema integrado como módulo único enfrentado al diccionario completo), y cómo se produce una saturación de la curva global, lo que nos permitirá recortar convenientemente el tiempo de proceso, de acuerdo con la discusión del Apartado 3.4.2. Un valor de longitud de lista de preselección del 10% del tamaño del diccionario parece ser un valor razonable, con pérdidas mínimas de tasa conjunta respecto al máximo obtenible.

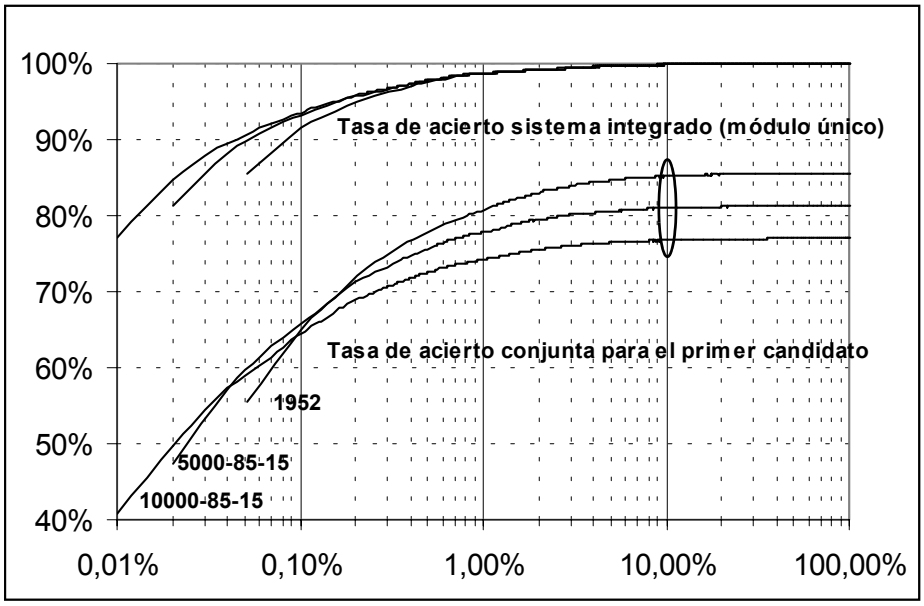

Figura 3-23: Tasa de acierto en función del número de candidatos usados para el sistema integrado como módulo único (gráficas superiores). Tasa de acierto para el primer candidato en el sistema en dos pasos sobre VESTEL-L (gráficas inferiores), para los tres diccionarios definidos. Modelado semicontinuo independiente y dependiente del contexto con `alf45`.

El cruce de las curvas se debe, como se ha comentado en otra ocasión, al uso de un eje de abscisas normalizado respecto a la longitud del diccionario. Evidentemente, las tasas de reconocimiento obtenidas son mejores cuanto menor es el tamaño del diccionario.

3.6 Conclusiones

En este capítulo se ha analizado el comportamiento de distintas alternativas arquitecturales en el diseño de sistemas de reconocimiento. Se han tratado por un lado los enfoques integrados y no integrados, así como los basados en el paradigma de hipótesis-verificación, como estrategia de diseño de sistemas con el objetivo de reducir la carga computacional necesaria, tratando de mantener en lo posible las tasas de reconocimiento globales. Se han descrito igualmente las implementaciones particulares realizadas en esta tesis como muestra de cada uno de los enfoques arquitecturales.

Se ha hecho un estudio teórico de la formulación a plantear de cara al diseño de sistemas multi-módulo, tanto en tiempo de proceso (coste computacional) como en tasa conjunta de reconocimiento, llegando al establecimiento de una metodología de diseño de los mismos y ofreciendo las pautas necesarias para determinar la adecuación de módulos particulares de cara a su integración en un sistema multi-módulo. Como muestra de aplicación de dicha metodología, se ha trabajado sobre un caso real concreto, estableciendo las condiciones de adecuación necesarias para asegurar los objetivos sobre una tarea concreta, todo ello sin necesidad de realizar la integración de los módulos disponibles y haciendo una mínima experimentación, mucho menos costosa que la que requeriría el cálculo completo y preciso con el sistema final.

Se ha establecido la absoluta conveniencia del diseño de reconocedores multi-etapa, dada la posibilidad de conseguir drásticas reducciones en tiempo de proceso manteniendo tasas de reconocimiento muy próximas a los máximos alcanzables.

En lo que respecta al comportamiento de la tasa conjunta en sistemas multi-etapa, se ha mostrado los límites obtenibles de tasa de error, estableciendo la importancia de contar, no solamente con módulos de hipótesis que aseguren con alta probabilidad la inclusión de la palabra a reconocer en la lista de preselección, sino también con módulos de verificación que deben ser capaces, por sí solos, de enfrentarse a la tarea completa (sin reducción de diccionario) y obtener la tasa deseada para el sistema conjunto. La evaluación se ha realizado también experimentalmente, obteniendo las curvas de tasa conjunta en función de la longitud de lista de preselección usada.

Se han presentado igualmente los resultados obtenidos por las arquitecturas implementadas sobre dos tareas radicalmente distintas (habla limpia dependiente del locutor y habla telefónica independiente del locutor), estableciendo la importancia del uso de un sistema integrado que permita un guiado explícito en la búsqueda acústica, desde el principio, en las tareas acústicamente más complejas. En la evaluación de los módulos de hipótesis se ha mostrado cómo es factible conseguir tasas de inclusión inferiores al 2% para un tamaño de lista de preselección inferior al 10% del tamaño del diccionario sobre la tarea VESTEL-L (lo que permite conseguir un 85% de ahorro de tiempo de proceso respecto al sistema en un único paso), bajando hasta tan sólo 10 candidatos para la tarea POLYGLOT, dada su menor complejidad acústica (con lo que en este caso, el sistema no integrado casi es capaz de enfrentarse por sí solo a dicha tarea, sin más que incrementar ligeramente la potencia del modelado acústico utilizado).

Adicionalmente, se han propuesto dos estrategias de medida y evaluación especialmente orientadas a nuestros intereses: la primera, consistente en medir las curvas de tasa de error de inclusión en función, no del número de candidatos seleccionados, sino del porcentaje que dicho número implica referido al tamaño del diccionario usado en la tarea, como mecanismo más conveniente para efectuar comparaciones; la segunda, una medida alternativa de disminución relativa de tasa de error en función de la tasa de error del sistema base, lo que puede permitir obtener una visión más rica del efecto de distintas estrategias en sistemas con tasas base muy distintas.

4 Reducción del espacio de búsqueda

4.1 Introducción

Los requisitos computacionales son uno de los factores principales a tener en cuenta a la hora de acometer el diseño de sistemas pensados para operar en tiempo real, especialmente cuando hablamos de sistemas de información sobre línea telefónica.

Los operadores de dichos sistemas demandan sistemas y algoritmos que les permitan incrementar el número de reconocedores activos que pueden correr sobre un soporte físico concreto, de forma que aumente el número de canales a los que pueden dar servicio y disminuir así los costes de producción y operación.

Eso implica fundamentalmente disminuir los tiempos de proceso sin perjudicar la tasa de reconocimiento del sistema, aunque, en ocasiones, pueden buscarse soluciones de compromiso que balanceen de forma adecuada ambos factores: ahorro en tiempo de proceso y pérdida de precisión en el reconocimiento. En cualquier caso, los sistemas del mundo real deben asegurar tasas muy cercanas al 100% para que sean aceptados por los usuarios, lo que se traduce en severas condiciones de funcionamiento para aquellos.

Los sistemas de reconocimiento automático de habla, al estar basados en complejas técnicas de reconocimiento estadístico de patrones, son caros computacionalmente, de forma que en la literatura encontramos multitud de aproximaciones a la hora de reducir el esfuerzo en el proceso de búsqueda o para mejorar la eficiencia de los algoritmos usados.

En este capítulo partiremos de un análisis previo de complejidad y demanda computacional, para centrarnos en primer lugar en las alternativas de exploración y búsqueda consideradas en esta tesis, con las correspondientes consideraciones experimentales. El grueso del capítulo lo constituye el trabajo desarrollado alrededor de la idea de listas de preselección de longitud variable, finalizando con la aplicación de ese enfoque a sistemas de estimación de la fiabilidad de las resultados propuestos por los reconocedores. Un apartado de conclusiones particulares cierra el capítulo.

4.2 Análisis previo de complejidad y demanda computacional

La medida de complejidad y requisitos computacionales de sistemas suelen ser sumamente complicada, sobre todo cuando se trata de analizar distintas estrategias a través de las comparaciones.

En el terreno teórico, puede ser factible hacer una estimación analítica de coste computacional y determinar el orden de complejidad de algunos algoritmos, incluso acercándonos más a estimaciones sobre la implementación dada [Macías92]. Sin embargo, dicha estimación teórica puede o no estar directamente relacionada con los requisitos de una implementación real en la que multitud de factores adicionales deberían ser tenidos en cuenta (pericia del programador en la codificación, capacidad de optimización del compilador, arquitectura software y hardware del sistema, etc.).

En la práctica, y será el enfoque que usemos en este trabajo, asumiremos que la implementación disponible de cada uno de los sistemas/módulos analizados está en un nivel de optimización similar (salvo indicación expresa), con lo que atenderemos a los tiempo de ejecución medios. Por supuesto todos ellos han sido obtenidos sobre la misma máquina y en condiciones similares de carga.

Tras medir los tiempos de cada uno de los módulos algorítmicos implicados (parametrización, cuantificación vectorial (modelos discretos), generación de cadena fonética (modelos discretos y semicontinuos independientes del contexto (IC), en función del alfabeto utilizado), acceso léxico (dependiente del alfabeto y del diccionario, compilación lineal o en árbol), algoritmo de Viterbi

(modelos discretos y semicontinuos, IC o DC, dependiente del diccionario y del alfabeto) y algoritmo de un paso guiado por árbol (dependiente del alfabeto y del diccionario, modelos discretos o semicontinuos IC), además de los de ordenación de candidatos), se verificó que los más caros computacionalmente son, aquellos en los que se hace la búsqueda sobre las estructuras que modelan los diccionarios, recorriendo el espacio asociado a la aquellas, con lo que son en ellos en los que nos centraremos en este capítulo. En la Tabla 4-1 se muestran los tiempos de proceso medio por palabra para distintos módulos de las implementaciones realizadas y distintos diccionarios. No entraremos en detalle en los mismos, salvo para verificar lo dicho anteriormente: *Acclex* y *Viterbi* son los módulos que realizan el acceso léxico y el proceso de reconocimiento del algoritmo de Viterbi.

Tabla 4-1: Tiempos de proceso medios por palabra (usando modelado semicontinuo con el alfabeto `alf45` en módulos de preselección (hipótesis) y verificación).

	Preselección (one-pass y acceso léxico)				***Verificación***		
Tarea	***Calcfactors[1]***	***OPSC[1]***	***AccLex[1]***	***Ordena[1]***	***MatrizB[1]***	***Viterbi[1]***	***Ordena[1]***
1952	*19'2ms* *25%*	*32'5ms* *42%*	*23'9ms* *31%*	*1'1ms* *2%*	*192'3ms* *18%*	*899'6ms* *82%*	*1'1ms* *0%*
5000-85-15	*19'2ms* *16%*	*31'9ms* *27%*	*64'9ms* *54%*	*3'4ms* *3%*	*199'6ms* *8%*	*2419ms* *92%*	*3'4ms* *0%*
10000-85-15	*19'5ms* *10%*	*33'2ms* *18%*	*129ms* *68%*	*7'4ms* *4%*	*205'4ms* *4%*	*5171ms* *96%*	*7'4ms* *0%*

1. Se incluyen tiempos para fases particulares del proceso (Calcfactors: calcula las probabilidades extraídas de los modelos semicontinuos para realizar el reconocimiento; OPSC: algoritmo de un paso; AccLex: algoritmo de acceso léxico que recorre todo el diccionario; Ordena: ordenaciones de candidatos; MatrizB: cálculo de los valores de probabilidad a usar en el proceso de viterbi posterior; Viterbi: algoritmo de Viterbi)

4.3 Estrategias de exploración/búsqueda

La optimización de la búsqueda es fundamental en los sistemas modernos de procesamiento de habla y lenguaje natural [Cole95]. Los métodos óptimos de búsqueda son aquellos que encuentran la solución óptima (o una de las mejores, si hay varias), pero en ellos hay que tener en cuenta, además, consideraciones de velocidad y uso de recursos computacionales (además de otros atributos como pueden ser el retardo en producir la respuesta y el proporcionar información adicional, como múltiples candidatos, valores de coste o probabilidad, etc.). Los métodos que han conseguido un mejor comportamiento han sido los basados en programación dinámica [Bellman57].

La mayoría de los sistemas de reconocimiento de gran vocabulario disponibles en la actualidad cuentan con mecanismos de compresión estática y/o dinámica de su espacio de búsqueda, como medio fundamental de reducir la elevada carga computacional asociada.

Los mecanismos de compresión estática se basan en el uso de estructuras de grafos con o sin restricciones en su construcción, que sirven para almacenar información para el guiado de algoritmos de programación dinámica sobre espacios acústicos o léxicos (diccionarios) [Macías96i], gramatical (gramáticas basadas en autómatas) [Colás99] e incluso semántica [Levin95]:

- Grafos genéricos: Sin ninguna restricción a priori, en la que cada nodo puede estar conectado con cualquiera. Un ejemplo de grafo genérico no determinista es el mostrado en la Figura 4-1, donde en cada nodo aparece el símbolo fonético asociado y los números de índice de las palabras asociadas, para el diccionario compuesto por casa (0), cosa (1), cesa (2) y cocina (3).

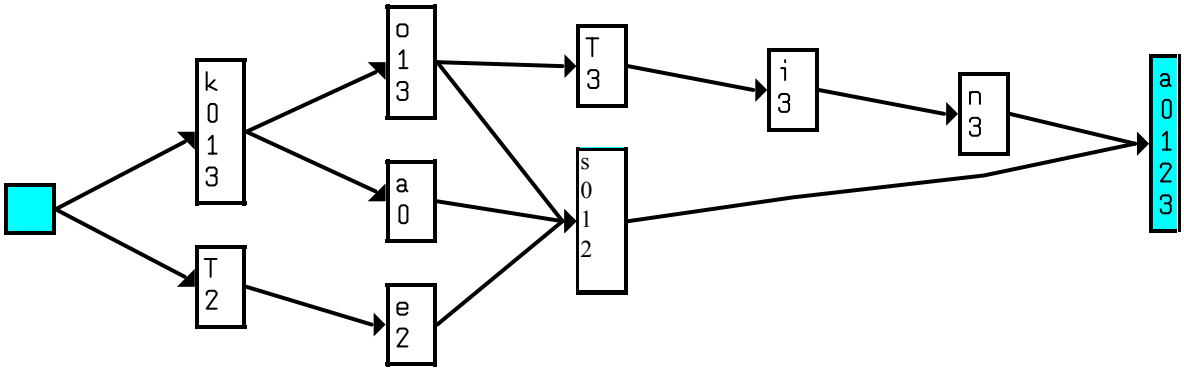

Figura 4-1: Ejemplo de estructura de grafo genérico para las palabras casa, cosa, cesa y cocina.

- Grafos deterministas: Que permiten recuperar de forma unívoca el camino óptimo resultado de la búsqueda. Para ello es necesario incluir información adicional en los nodos, marcando aquellos que caracterizan unívocamente cada una de las posibles secuencias almacenadas. Un ejemplo de grafo determinista es el mostrado en la Figura 4-2, donde los nodos identificados como deterministas se han marcado en verde.

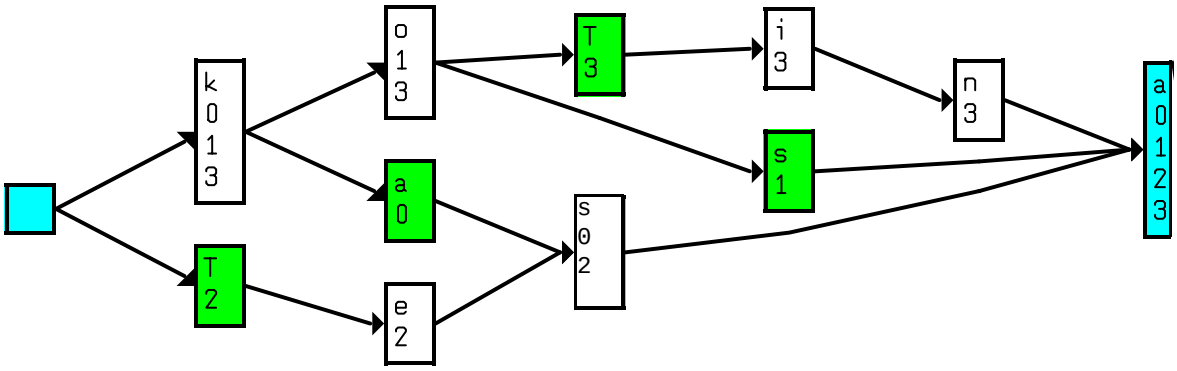

Figura 4-2: Ejemplo de estructura de grafo determinista para las palabras casa, cosa, cesa y cocina.

- Árboles: Cuya restricción fundamental es que un nodo sólo puede tener un predecesor. Esta característica los hace sumamente eficientes como estructuras de guiado en sistemas de reconocimiento, simplificando notablemente los cálculos a realizar en algoritmos de programación dinámica y proporcionando ahorros considerables en el espacio de búsqueda, como se verá posteriormente. Un ejemplo de estructura de árbol es el mostrado en la Figura 4-3.

4.3.1 Algoritmos de programación dinámica y estructuras de búsqueda

Las técnicas más utilizadas en sistemas de reconocimiento automático de habla están basadas en los principios de programación dinámica [Bellman57]. Dichos principios son fundamentales a la hora de plantear la búsqueda de un camino óptimo a través de un espacio de búsqueda dado, implementado sobre una estructura de grafo genérico, en el que los caminos se evalúan a través de la acumulación de valores de coste a lo largo de ellos.

Los algoritmos de programación dinámica toman decisiones locales en cada punto del espacio de búsqueda (nodo del grafo) y las estructuras de búsqueda basadas en grafos (genéricos o específicos, como los árboles) están especialmente bien adaptados a servir de guiado a dichos algoritmos.

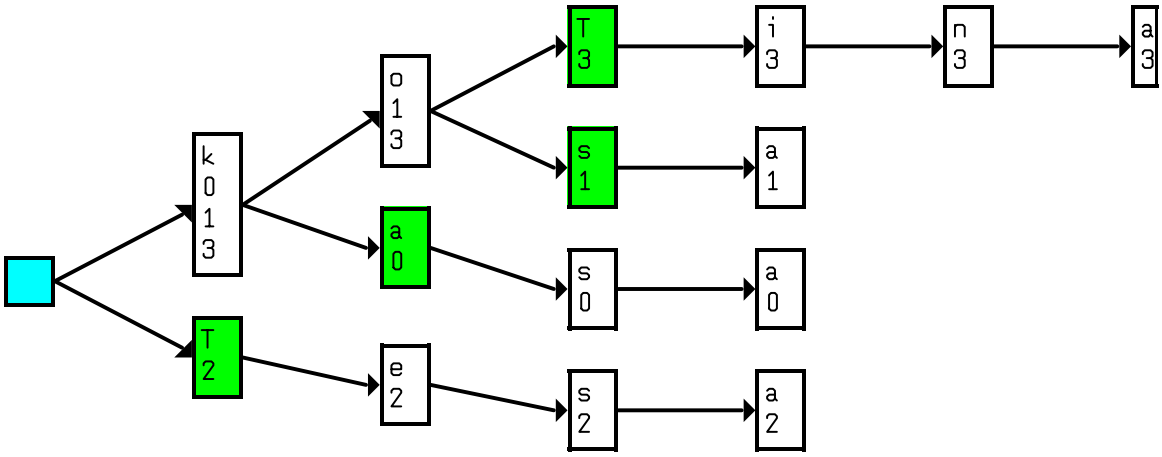

Figura 4-3: Ejemplo de estructura de árbol para las palabras casa, cosa, cesa y cocina.

4.3.2 Búsqueda sobre grafos genéricos

La búsqueda sobre gráficos genéricos presenta serios problemas de implementación dada la necesidad de guardar la historia anterior de forma completa, para ser capaces de hacer una búsqueda hacia atrás (*backtracking)* consistente que nos permita recuperar la secuencia de símbolos correcta.

El proceso de construcción de grafos genéricos a partir de un vocabulario dado se encarga de unificar tanto las cabeceras como las colas de las palabras de aquel, comprimiendo muy significativamente el espacio de búsqueda y, en consecuencia, reduciendo drásticamente el tiempo de proceso necesario.

4.3.3 Búsqueda sobre grafos deterministas

Un grafo determinista se diferencia de un grafo genérico en la imposición de un criterio adicional: Durante la construcción del grafo se asegura la existencia de nodos especiales que identifican unívocamente una única palabra del vocabulario.

Esa prevención en la construcción ayuda fundamentalmente al proceso de identificación de la producción de habla realizada, de forma que cada palabra tiene un nodo asociado característico que nos permitiría conocer de cuál se trata. Sin embargo, también se realiza una unificación de colas, con lo que un nodo final puede caracterizar a más de una palabra, de modo que todas las asociadas a él tendrán asociado el mismo coste (la misma probabilidad) y se complican las posibilidades de conseguir una ordenación eficiente, al haber *empates* constantes en la lista de palabras ordenadas propuestas.

4.3.4 Búsqueda sobre árboles

Un árbol es un grafo en el que se ha impuesto la condición de que un nodo sólo puede tener un antecesor. El proceso de construcción se encarga de agrupar las historias comunes de símbolos en los comienzos de las palabras del vocabulario, por lo que se produce una compresión importante del espacio de búsqueda, al no tener que repetir cálculos adicionales para cada una de ellas, si bien dicha compresión no es tan grande como en el caso de los grafos.

4.3.5 Consideraciones de ahorro en tiempo de proceso

A la hora de estudiar el impacto del uso de distintas estructuras del espacio de búsqueda en el comportamiento e implementación del sistema hay que considerar factores de espacio ocupado en

memoria y de tiempo de proceso, además de medir cuantitativamente las tasas obtenidas con cada uno de ellos. En este apartado describiremos los dos primeros factores, dejando para el Apartado 4.3.6 el tercero.

En la Tabla 4-2 se muestran los valores del número de nodos a utilizar para cada uno de los espacios de búsqueda estudiados y cada uno de los diccionarios usados en la mayor parte de las tareas con las que nos hemos enfrentado[1], en la que además se incluye el factor de incremento que supone en número de nodos con respecto al del diccionario de 1175 palabras. El número de nodos está directamente relacionado con el tamaño de memoria ocupado, dependiendo de forma lineal de aquél.

Tabla 4-2: Número de nodos en función de la estructura del espacio de búsqueda utilizado

DICC.	*Factor*	*Nodos búsqueda lineal*	*Factor*	*Nodos búsqueda en árbol*	*Factor*	*Nodos búsqueda grafo determinista*	*Factor*	*Nodos búsqueda grafo no determinista*	*Factor*
1175		7120		3926		2316		1639	
1996	1,7	15011	2,1	7942	2,0	4532	2,0	3332	2,0
5000DV	4,3	34195	4,8	17847	4,5	9259	4,0	5882	3,6
5000IV	4,3	44951	6,3	24112	6,1	12218	5,3	9134	5,6
10000DV	8,5	69420	9,8	32960	8,4	17107	7,4	9365	5,7
10000IV	8,5	94884	13,3	48617	12,4	21516	9,3	14584	8,9

Como puede observarse, un incremento en el número de palabras del diccionario implica un aumento del número de nodos en todos los espacios de búsqueda. Sin embargo, el impacto de dicho incremento es menor cuanto más compacta es la estructura usada (cuanto más compartición de elementos permite). Es importante indicar que estos resultados deben analizarse también a la luz de los datos de longitudes medias de diccionarios, que se ofrecen en el Apartado C.4 del Anexo C, a partir de la página 198.

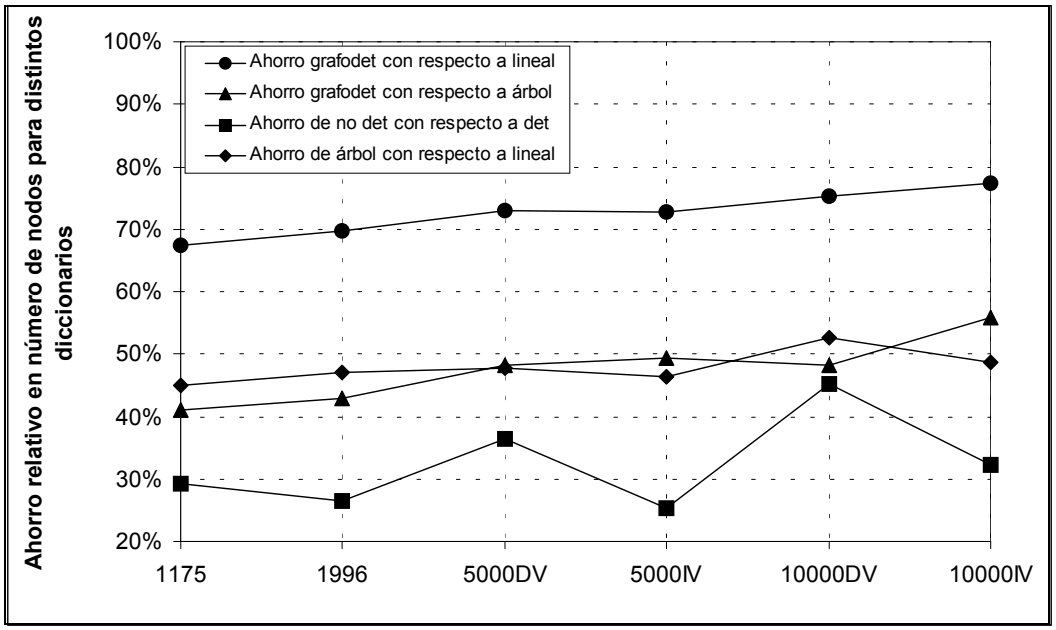

Figura 4-4: Ahorro relativo en número de nodos para distintas estructuras del espacio de búsqueda

Evidentemente, la implementación en estructuras de grafo proporciona ahorros muy importantes en espacio ocupado, tal y como muestra la Figura 4-4. La estructura de árbol produce una

1. La composición de dichos diccionarios se describe en detalle en el Anexo B.2.3 a partir de la página 190.

reducción que ronda el 50% del espacio de búsqueda, al compararla con la búsqueda lineal. Dicha reducción ronda el 70% si se trata del grafo determinista.

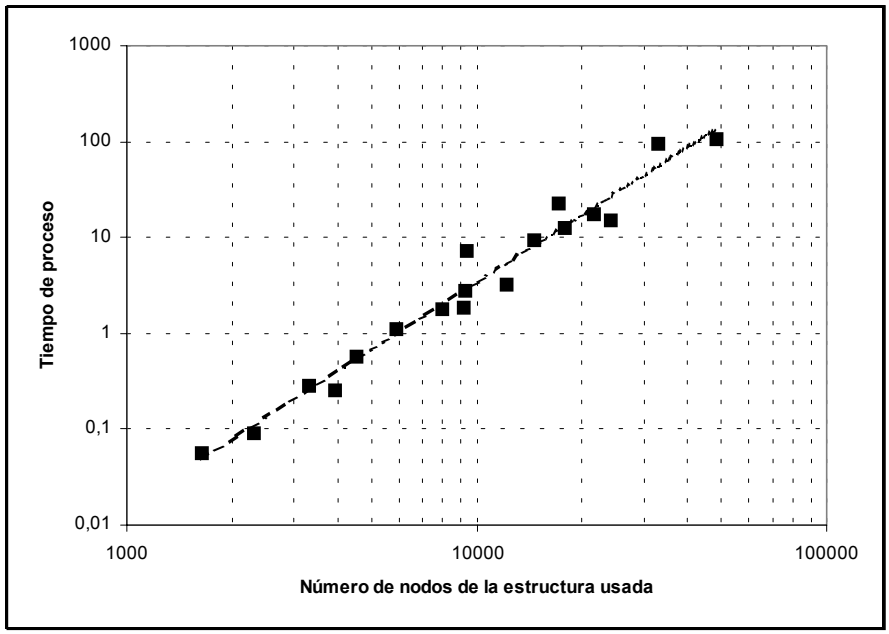

Figura 4-5: Tiempo de proceso en función del número de nodos

Si atendemos a los tiempos de proceso del algoritmo de búsqueda, obtenemos los resultados mostrados en la Figura 4-5, donde puede observarse la línea de tendencia superpuesta, que es una función potencial (los ejes de coordenadas es logarítmico) del número de nodos de cada estructura de búsqueda, habiéndose determinado una relación del tipo $\text{tiempo} \cong k \cdot N^2$ (el exponente de la línea de tendencia de la figura es exactamente 2'3409). Los ahorros en tiempo de proceso para cada diccionario se incluyen en la Figura 4-6.

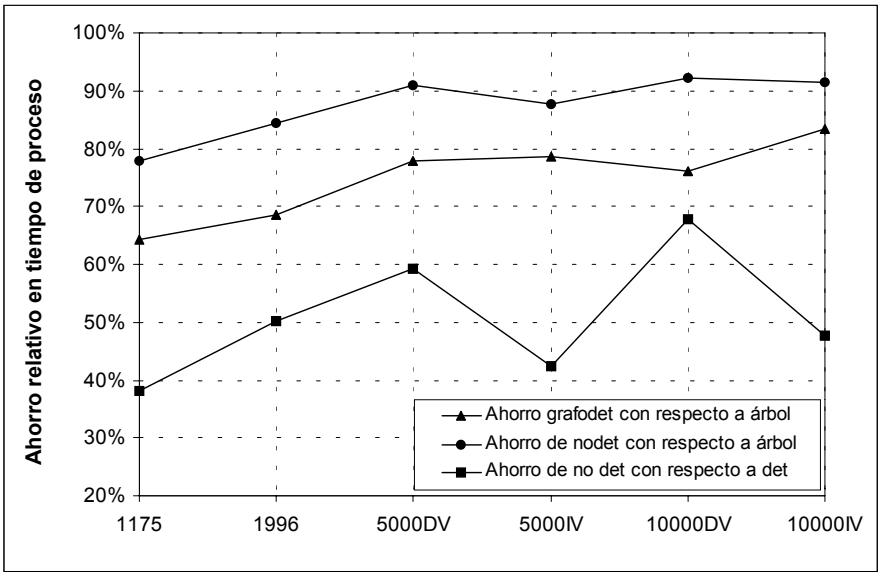

Figura 4-6: Ahorro relativo de tiempo de proceso para cada diccionario

En esta tesis se hizo un estudio detallado de la relación entre distintas medidas de complejidad del grafo y el tiempo de proceso, y se llegó a estimar con razonable precisión que el parámetro más relevante de estimación dentro de la misma estructura de búsqueda, es el número de nodos finales[1] de la misma, como puede observarse en la Figura 4-7, donde las líneas de tendencia son también potenciales de orden próximo a 2.

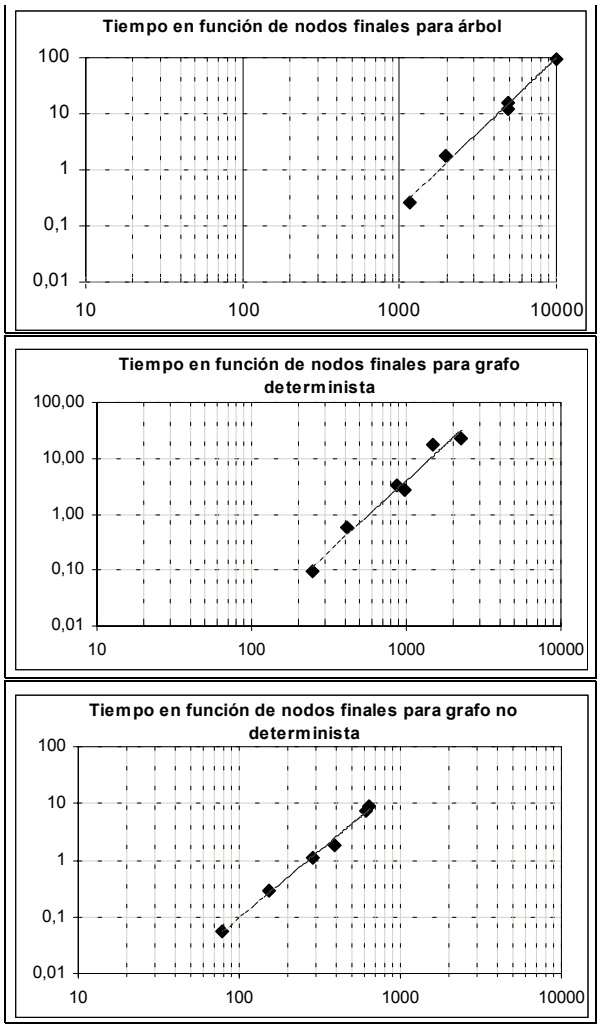

Figura 4-7: Tiempo de proceso en función del número de nodos finales para cada estructura de búsqueda

A partir de la discusión y los valores obtenidos en los cuadros y gráficas de este apartado, está claro que las mayores optimizaciones por compresión del espacio de búsqueda las conseguimos con el grafo no determinista, siguiéndole muy de cerca en capacidad de optimización el grafo determinista (lo que es muy importante, dadas las ventajas en cuanto a identificación que proporciona) y, finalmente, el árbol.

El grafo determinista sería nuestra propuesta de estructura de búsqueda si contamos con un modelo acústico lo suficientemente potente, ya que tendremos la capacidad de identificar con precisión la palabra de máxima probabilidad. Sin embargo, también habría que estudiar el impacto de la no optimalidad de la solución propuesta al usar la estructura de grafo, en lugar de la de árbol (o la lineal).

4.3.6 Consideraciones sobre la tasa de inclusión

La diferencia fundamental entre estructuras de búsqueda basadas en árboles y en grafos genéricos radican, como se ha visto, en la mayor o menor compresión del mismo. El mayor problema derivado de la compresión en un grafo radica en la unión de nodos finales, si lo comparamos con la estructura en árbol: un nodo final estará asociado a más de una palabra. Esta unión implica que, cuando el algoritmo de programación dinámica acaba de realizar su proceso de cálculo, asignará el mismo coste

1. La línea de tendencia con el número de nodos totales también presenta una alta correlación con las medidas reales, pero es ligeramente menor que la obtenida para los nodos finales.

a todas las palabras asociadas a cada nodo final. De esta forma, y si no abordamos soluciones a esta situación, seremos incapaces de decidir qué palabras dentro de cada nodo tienen más o menos probabilidad de corresponder a la producción de habla realizada, aunque seamos capaces de decidir cuál es la palabra más probable.

El problema se agrava cuantas más palabras, en media, haya en cada nodo final, lógicamente. En la Figura 4-8 se muestra un ejemplo de la curva de tasa de error de inclusión al usar una estructura de árbol y una de grafo en el algoritmo de acceso léxico, con un vocabulario de 1175 palabras. Como puede observarse, los resultados al usar el grafo son totalmente inadmisibles para su inclusión en un sistema real: no conseguimos altas tasas de inclusión salvo para valores muy elevados de longitud de lista de preselección[1].

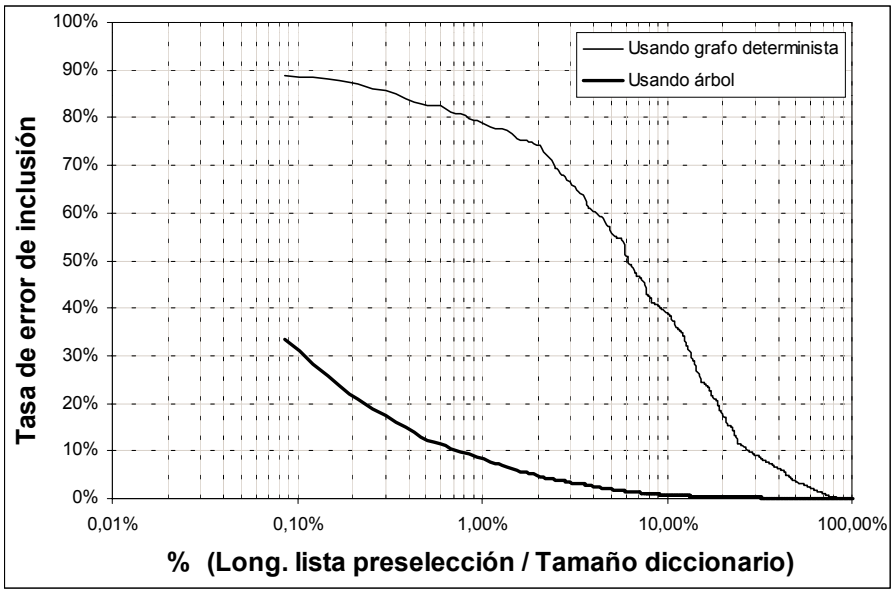

Figura 4-8: Comparativa de tasa de error de inclusión al usar una estructura de árbol y una de grafo determinista.

Sin embargo, las importantes reducciones de tiempo de proceso obtenibles por las estructuras de grafo, nos animaron a realizar un estudio de la posible inclusión de mecanismos adicionales de corrección de las probabilidades asignadas a cada palabra dentro de un mismo nodo final, con el objetivo de incrementar la tasa global de inclusión y llegar a valores que nos permitieran competir con las tasas obtenidas al usar estructuras de árbol.

En la experimentación que se desarrolló se tomó como sistema de trabajo el módulo de acceso léxico, por su simplicidad de tratamiento. Sin embargo, sería posible analizar el efecto sobre cualquiera de los sistemas diseñados en los que las estructuras de árbol o grafo son aplicables, es decir, cualquiera que permita usar un espacio de búsqueda guiado por dichas estructuras en el proceso de optimización (programación dinámica).

El elemento de decisión usado en el algoritmo de acceso léxico con estructura de búsqueda lineal o basada en árbol (que son equivalentes desde el punto de vista de resultado final) es el coste de alineamiento (coste de acceso léxico *costeAL*). Tras analizar el problema, se detectaron una serie de parámetros (longitud de la cadena fonética, distribución de las longitudes de las palabras de cada nodo final, secuencia de palabras observadas durante el *backtracking*, etc.) que serían candidatos a colaborar en una nueva estimación del coste a usar, en combinación con el proporcionado por el acceso léxico

1. En la generación de listas de palabras propuestas se han seguido el orden de las mismas tal y como están anotadas en los nodos finales, de modo que dos palabras asociadas a un mismo nodo final no se consideran como reconocidas en la misma posición, sino una después de la otra. De ahí que la curva de tasa de error de inclusión no tenga *saltos* bruscos.

($costeAL_i$)[1]. De todos ellos, se decidió que el más prometedor era precisamente el relacionado con la secuencia de palabras observada con el *backtracking*: si se incluye un etiquetado adecuado de los nodos del grafo, marcándolos con el identificador de todas las palabras asociadas a cada uno, es factible calcular el número de veces que cada palabra aparece asociada a los nodos de la cadena fonética reconocida y utilizar dicho valor para modificar adecuadamente el coste asociado a cada palabra.

Así, nos fijaremos en dos valores:

- El coste de acceso léxico asociado al nodo final al que pertenece cada palabra
- Un coste adicional basado en la distribución del número de ocurrencias de una palabra en el camino calculado por el backtracking.

Si pensamos en este último valor, el cálculo que nos interesa realizar para cada palabra *i*, corresponde a una variable aleatoria (que llamaremos *ocupación*) que debería ser dependiente de la diferencia entre el número de veces que aparece la palabra en los nodos del camino óptimo (que llamaremos *numOcurrencias*) y la longitud del mismo (que llamaremos *longitudCamino*). El camino óptimo será el calculado a partir del nodo final asociado a la palabra *i*, *nodoFinal(i)*:

$$ocupacion_i = numOcurrencias_i - longitudCamino_{nodoFinal(i)}$$

así, el mejor caso debería darse para un valor 0, es decir, la palabra dada ha aparecido en todos los nodos del camino óptimo.

Asumiremos que esa variable aleatoria sigue una distribución normal, de modo que, en cada caso, asociaremos un valor de probabilidad igual a:

$$p(ocupacion) = \frac{1}{\sqrt{2\Pi}\sigma} e^{-\frac{ocupacion^2}{2\sigma^2}}$$

desarrollando las fórmulas vistas tomando logaritmos para convertir las probabilidades en costes, podemos llegar a la siguiente expresión final del coste ponderado de cada palabra (en la que hemos prescindido de la inclusión de constantes que no afectan al resultado final):

$$costePonderado_i = \alpha \cdot \frac{1}{\sigma^2_{costeOcupacion}} \cdot \text{costeOcupacion}_i + (1-\alpha) \cdot \frac{1}{\sigma^2_{costeAL}} \text{costeAL}_i$$

en la que aparecen como factores de ponderación el factor α y las varianzas correspondientes a las distribuciones de los costes de ocupación ($costeOcupacion_i$=$ocupacion_i^2$) y los de acceso léxico.

En la experimentación llevada a cabo se analizaron previamente las variaciones producidas al usar distintas estrategias en el cálculo de las varianzas: globales (calculadas sobre todas las palabras del grafo), dependientes del nodo final asociado (calculadas sobre las palabras asociadas a cada nodo final) e independientes (valores fijos en un rango dado). Los mejores resultados se obtuvieron al usar varianzas globales, dado que las varianzas dependientes de nodos finales tenían un grave problema de estimación para nodos con un número bajo de palabras asociadas (incluso aplicando técnicas de suavizado).

Para facilitar el proceso de evaluación se desarrollaron una serie de medidas comparativas entre el rendimiento del sistema usando la ordenación basada en costes de acceso léxico únicamente y la ponderada, basadas en distintos criterios:

- Número absoluto de *palabras* en las que funcionaba mejor el acceso léxico con costes ponderados o sin ellos.

1. En la búsqueda realizada por el módulo de acceso a léxico sobre árbol, el único elemento que contribuye a la ordenación es el coste de acceso léxico asociado, calculado al comparar la cadena fonética de entrada con todas las palabras del diccionario

- Suma absoluta de *posiciones de acierto* (medidas como posiciones de la lista de preselección en las que se reconoció cada palabra) para la estrategia que funcionaba mejor para cada palabra, usando o no costes ponderados
- Suma absoluta de *posiciones de mejora* (medidas como diferencias de posiciones de acierto) para la estrategia que funcionaba mejor en cada caso, usando o no costes ponderados
- Suma absoluta de *momentos de mejora* (medidos como el producto de la diferencia en posiciones de mejora y la posición reconocida) para la estrategia que funcionaba mejor en cada caso, usando o no costes ponderados
- Diferencias entre los valores anteriores calculados para costes ponderados y los calculados para costes no ponderados

La idea que perseguíamos era buscar una posible correlación entre dichos valores y las curvas de preselección que mostraban un mejor comportamiento, para tener un criterio objetivo de decisión acerca del valor de α a usar. La calificación de *mejor comportamiento* se hizo usando, entre otros factores, indicadores como el de posición en la que se alcanzaban determinadas tasas de inclusión, como por ejemplo la mostrada en la Figura 4-9, en la que se muestra la posición del candidato en el que se alcanzaba la tasa de inclusión del 97%, 98% y 99%, medida como porcentaje sobre el tamaño del diccionario.

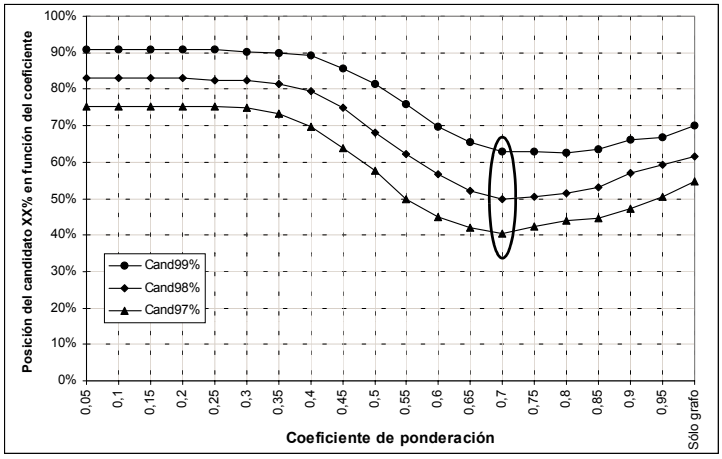

Figura 4-9: Tamaño de lista de preselección necesario (medido como porcentaje sobre el tamaño del diccionario) para obtener tasas de inclusión del 97%, 98% y 99%, en función del valor del coeficiente de ponderación α y para el caso del uso de grafo sin reordenar.

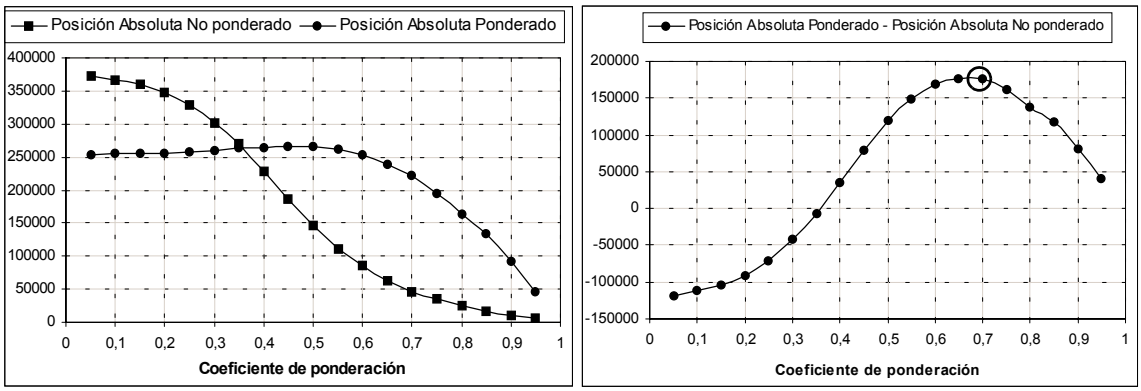

Figura 4-10: Curvas de medida de suma de posiciones absolutas usando o no costes ponderados (izquierda) y diferencia entre ambas (derecha), en función del coeficiente de ponderación.

En la Figura 4-10 se muestran como ejemplo las curvas de suma absoluta de posiciones de acierto (gráfica de la izquierda, que se incluye como referencia) y la diferencia entre ambas (gráfica de la derecha) para el acceso léxico usando costes no ponderados (en los que la ordenación de candidatos tiene únicamente en cuenta el valor del coste de acceso léxico obtenido) y ponderados (en los que la ordenación se hace de acuerdo con la formulación vista anteriormente). La mejor correlación entre los criterios y la bondad de las curvas de preselección se obtuvo para las basadas en diferencias de parámetros, con la particularidad de conseguir un acuerdo bastante uniforme entre todas ellas en cuanto al valor óptimo de α. En el ejemplo, se optó por usar un valor de $\alpha = 0'7$.

En la gráfica izquierda de la Figura 4-11 se muestra una comparativa de las curvas de tasa de

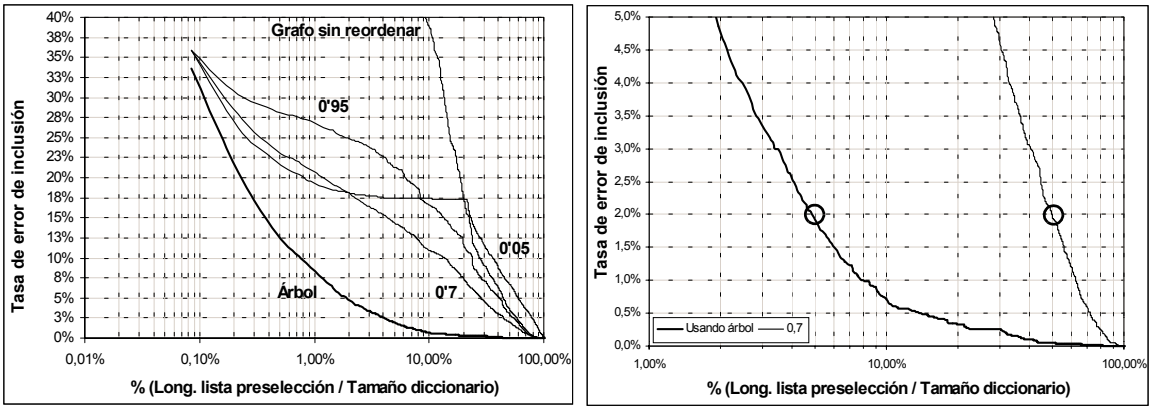

Figura 4-11: Izquierda: Curvas de tasa de error de inclusión comparando el uso de costes no ponderados (Grafo sin reordenar), ponderados con distintos valores de ponderación (0'05, 0'95 y 0'7) y usando un árbol. Derecha: Detalle de la curva de inclusión para la ponderación seleccionada de 0'7 en comparación con el árbol.

error de inclusión para distintos valores de ponderación, el no usarla y el caso de referencia usando un árbol. Como puede observarse estamos aún lejos de conseguir llegar al funcionamiento de éste último, pero hemos mejorado muy significativamente comparando con el caso de usar un grafo sin mecanismos de ponderación de costes para reordenación.

En la parte derecha de la Figura 4-11 se muestra un detalle de la curva de tasa de error de inclusión con la ponderación seleccionada y la correspondiente al árbol. Sobre ella y la de medidas de ahorro relativo de tiempo de la Figura 4-6 se pueden hacer consideraciones de compromiso entre tasa y tiempo. En el caso del árbol, necesitamos un 4% de longitud de lista de preselección para llegar al 2% de tasa de error de inclusión[1], y en el caso del grafo con costes ponderados y factor 0'7, hay que subir hasta un 50%. Con esos números, el uso del grafo no es una alternativa competitiva con respecto al árbol, por lo que no abundamos más en esta aproximación, dejando para trabajos futuros la determinación de estrategias de ponderación más potentes.

4.4 Longitud de las listas de preselección

Como ya se ha comentado, en sistemas basados en el paradigma hipótesis verificación, es crucial que el módulo de análisis grueso asegure con alta probabilidad que la palabra efectivamente pronunciada está contenida en la lista de candidatos (lista de preselección) que se entregan al módulo de análisis detallado.

En el desarrollo de sistemas de este tipo, el diseñador tiene que tomar decisiones relativas a la potencia del modelado, por un lado, y al tamaño de dicha lista de preselección, por otro. Hay que establecer un compromiso entre ambos parámetros de diseño: cuanto más detallado y potente sea nuestro modelado acústico, menor será la lista de preselección necesaria para asegurar una tasa dada. Ambos factores influyen en la potencia computacional requerida, de forma que habrá que tener en

1. Siendo éste valor un criterio de diseño de nuestro sistema de verificación.

cuenta no sólo la reducción conseguida en la etapa de hipótesis, sino el impacto en tiempo de proceso que dicha reducción tiene sobre el tiempo total de ejecución.

En este apartado plantearemos diversas alternativas para seleccionar una longitud de lista adecuada, imponiendo como condición básica el mantenimiento de una tasa de error de inclusión máxima.

En general nuestro objetivo será conseguir bajar al máximo el tamaño de dicha lista, dado que en la etapa de verificación (análisis detallado) se utilizan típicamente algoritmos mucho más costosos computacionalmente, de modo que cuanto menor sea dicha lista, menor será la demanda de potencia de cálculo requerida. En general, esto es una tarea complicada, sobre todo cuando los modelos utilizados en la fase de preselección son poco detallados.

4.4.1 Planteamiento general

El esquema genérico que pondremos en práctica es el mostrado en la Figura 4-12, en la que un bloque decisor se encargará de indicarle al módulo de preselección (análisis grueso) la longitud de la lista que debe entregar al módulo de análisis detallado, usando para ello un cierto conjunto de variables extraíbles de los procesos de parametrización y análisis grueso.

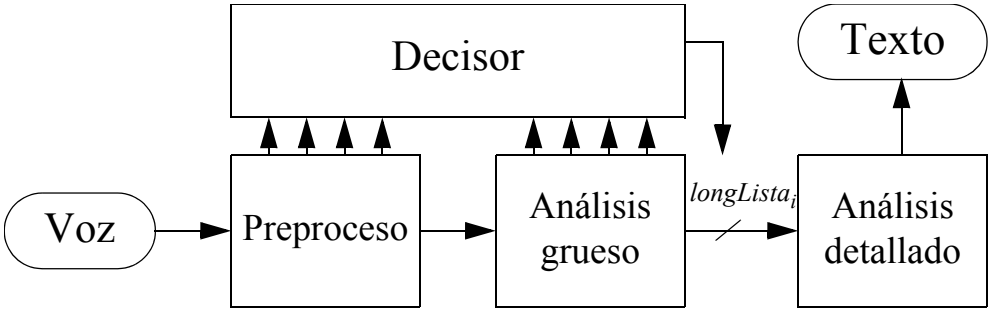

Figura 4-12: Esquema genérico de un SRAH basado en el paradigma hipótesis-verificación

Será sobre este bloque decisor sobre el que trabajaremos, aplicando distintas técnicas que describiremos con detalle en los apartados siguientes.

A medida que reduzcamos el coste computacional (minimización de la longitud de la lista de preselección), tendremos una disminución en la tasa de inclusión obtenida. Por el contrario, si fijamos la tasa de reconocimiento a obtener (que recordamos aquí que es el objetivo prioritario en nuestro sistema), puede que no consigamos la reducción computacional deseable.

4.4.2 Listas de tamaño fijo

El enfoque tradicional al problema de la estimación de la longitud, utiliza listas de tamaño fijo, estableciéndose éste a partir de los resultados obtenidos durante la experimentación, de forma que se cumpla algún requisito de diseño, que suele ser la obtención de una tasa de error de preselección determinada, o bien la reducción del esfuerzo necesario a un valor dado.

Así, según las definiciones vistas, los diseñadores utilizarían una longitud fija e independiente de la palabra a reconocer, lo que implica un esfuerzo medio constante igual a dicha longitud, es decir:

$$longLista(i) = longFija = esfuerzoMedio, \qquad \forall i \text{ (realización acústica)}$$

En general, y debido al desigual funcionamiento de los sistemas de reconocimiento en función de cada realización acústica, dicha longitud debe ser fijada a un valor elevado, lo que automáticamente se traduce en un esfuerzo medio elevado (también constante en este caso).

Si intentamos cuantificar el *exceso* de esfuerzo invertido, podríamos definir el *desperdicio*[1] como la diferencia entre la longitud de la lista de preselección usada para reconocer una palabra dada y la longitud mínima que hubiera bastado para que dicha palabra fuera correctamente reconocida: $desperdicio(i) = longLista(i) - posicOK(i)$. A partir de él podríamos calcular el desperdicio medio, extendiendo el cálculo a todas las palabras de la base de datos evaluada. Así, el uso de listas fijas implicará también un desperdicio medio elevado (calculable a posteriori si tenemos una base de datos etiquetada) y unos requisitos de potencia computacional posiblemente superiores a los deseados.

Para dar una idea más precisa de las magnitudes de las que estamos hablando, en la Tabla 4-3 se incluyen los cálculos de desperdicio medio, para las bases de datos de la tarea VESTEL (PRNOK, PERFDV y PEIV1000[2]), con los diccionarios de 10000 palabras y utilizando modelado semicontinuo independiente del contexto con el alfabeto `alf23`[3], para tasas de inclusión del 98% y 99%. En cada caso se incluye el valor de la longitud de lista fija óptimo (es decir, mínimo) para obtener cada tasa (*longFija*) y el desperdicio medio (abreviado como *dm*).

Tabla 4-3: Desperdicio medio usando una longitud de lista de preselección fija (para PRNOK, PERFDV y PEIV1000, con diccionarios de 10000 palabras y modelado semicontinuo independiente del contexto con el alfabeto `alf23`)

	Para 98%		***Para 99%***	
Lista	***longFija***	***dm***	***longFija***	***dm***
prnok	372	348	621	594
perfdv	1353	1260	2567	2457
peiv1000	916	863	2092	2023

Evidentemente dichas medidas no son aplicables a un caso real, en el sentido de que no es posible conocer con antelación la posición en la que será reconocida una palabra, pero lo que sí nos dan es una idea clara de las magnitudes de los ahorros máximos a los que podríamos llegar si consiguiéramos estimar de forma precisa la longitud de la lista a utilizar. Analizando la Tabla 4-3 puede verse que los ahorros conseguibles por palabra (desperdicio medio) están próximos a los valores de longitud fija de lista que tendríamos que utilizar para cada tasa, confirmando la idea de que prácticamente todo el esfuerzo se invierte en recuperar palabras reconocidas, por los motivos que sean, muy lejos de las primeras posiciones.

Este resultado es, por supuesto, consistente con las curvas de tasa de inclusión de los experimentos realizados. A modo de muestra, en la Tabla 4-4, ofrecemos los resultados para las primeras posiciones de las tres listas en los experimentos con 10000 palabras. La columna *longFija* indica el número de candidatos usados y *tasa* la tasa obtenida en cada punto. En dicha tabla puede

1. La denominación de *desperdicio* procede de la idea de que si la lista de preselección es más larga que la posición en la que se reconoció correctamente la palabra, estaremos *desperdiciando* un cierto esfuerzo computacional para procesar candidatos, además de estar perjudicando la tasa del módulo de verificación, que se tendrá que enfrentar a un mayor número de palabras entre las que discriminar.
2. Que recordamos se definen en el Anexo B.2.2, a partir de la página 189, al describir el contenido de la base de datos TIDAISL.
3. Que recordamos utiliza 25 unidades alofónicas seleccionadas manualmente (23 alófonos+2 unidades de ruido), tal y como se describe en el Anexo D.2.3 a partir de la página 208.

observarse que, como mínimo, prácticamente más del 60% de las palabras son reconocidas en ese rango tan limitado de la lista de preselección.

Tabla 4-4: Tasas de inclusión para las primeras posiciones de la curva de preselección con tamaños fijos (para PRNOK, PERFDV y PEIV1000, con diccionarios de 10000 palabras y modelado semicontinuo independiente del contexto con el alfabeto `alf23`)

	***PRNOK5TR* (5810 palabras)**	***PERFDV* (2502 palabras)**	***PEIV1000* (1434 palabras)**
longFija	**tasa**	**tasa**	**tasa**
1	46,95%	30,14%	42,47%
2	57,40%	38,17%	53,07%
3	62,91%	43,45%	57,18%
4	66,59%	47,24%	59,97%
5	69,24%	49,96%	62,48%
6	71,41%	52,44%	65,34%
7	72,99%	54,36%	66,67%
8	74,49%	56,24%	68,34%
9	75,65%	57,79%	69,32%
	76,57%	58,95%	70,22%

Nuestro objetivo, por tanto, es hacer una estimación de la longitud fija a utilizar lo más ajustada posible a la posición en la que se reconoció cada palabra.

4.4.3 Listas de tamaño variable

La idea que estamos considerando al hablar de listas de tamaño variable es estimar una longitud de lista de preselección diferente para cada realización acústica, de forma que podamos reducir el esfuerzo medio necesario en el proceso de reconocimiento. Si dicho esfuerzo medio estuviera por debajo de la longitud fija de la que hablábamos más arriba y además mantuviéramos la tasa de error por debajo del umbral requerido (en nuestro caso el 2%), este enfoque disminuiría, en media, los requisitos computacionales del sistema (aunque, por supuesto, la reducción final calculada deberá tener en cuenta el tiempo de proceso del módulo de análisis fino).

El origen de este enfoque parte de la observación empírica de la existencia de una aparente relación entre la tasa de reconocimiento (y por consiguiente la posición en la que se reconocía cada palabra) y alguno de los parámetros disponibles en el proceso. Nos estamos refiriendo en concreto a que, en nuestros experimentos preliminares, las palabras más largas parecían ser más fácilmente reconocibles que las cortas, apareciendo en posiciones más bajas de la lista de preselección. Así, parece razonable pensar en que para palabras largas podremos utilizar una lista de preselección más corta que para aquellas de menor longitud.

Cuando hablamos de estimación (en este caso de una longitud de lista de preselección), podemos plantear el uso de diversas técnicas con distintas posibilidades de éxito en función, entre otras cosas, del grado de conocimiento a priori que tengamos sobre la tarea y la cantidad de datos experimentales disponibles.

En nuestro caso, no existe ninguna formulación algorítmica explícita acerca de la relación que estamos buscando y, lo que es peor, ni siquiera hay ninguna evidencia soportada teóricamente de que exista dicha relación.

El objetivo que perseguimos será utilizar cualquiera de los parámetros disponibles durante el reconocimiento (en la etapa de análisis poco detallado) en el proceso de estimación de la longitud de la lista de preselección.

Nuestro estudio abordará la aplicación de métodos paramétricos y no paramétricos. De los primeros haremos algunas exploraciones que usan estimación lineal, y de los segundos, plantearemos el

cálculo de tablas de corte y nos centraremos en la aplicación de redes neuronales como elemento decisor.

4.4.4 Selección de parámetros de entrada

Una de las preguntas más importantes que se plantean a la hora de seleccionar el conjunto de parámetros a utilizar es si el problema será resoluble con ellos[1]. De acuerdo con la idea expuesta más arriba acerca del desconocimiento de si existe o no una cierta relación entre los parámetros disponibles y la salida deseada, la respuesta a dicha pregunta no está en absoluto clara.

En nuestro caso, tenemos ciertas evidencias acerca de la correlación entre tasa de inclusión y longitud de palabra[2], por ejemplo, pero de cara a enfrentarnos con el diseño de un sistema de estimación y fruto de nuestro desconocimiento explícito del problema, debemos plantear un espectro amplio de posibilidades.

Remitimos al lector al Anexo A a partir de la página 185, donde se detalla el inventario de parámetros de los que disponemos, aunque incluimos aquí un resumen cualitativo de los mismos. A grandes rasgos, la idea es utilizar todos los parámetros extraíbles directamente de la etapa de preselección, además de usar derivaciones o variaciones sobre ellos. En concreto, podemos distinguir tres grandes grupos:

- Parámetros directos: Directamente obtenibles de datos de la ocurrencia acústica a reconocer o del proceso de preselección: número de tramas, longitud de la cadena fonética, coste del algoritmo de búsqueda acústica, número de símbolos en el diccionario del primer candidato reconocido en el acceso léxico, coste del acceso léxico para dicho candidato.
- Parámetros derivados: A partir de los anteriores, aplicando normalizaciones de distinto tipo (dividiendo por el número de tramas, por la longitud de cadena fonética, etc.): coste acústico normalizado por la longitud de palabra o de cadena; coste del acceso léxico para el primer candidato normalizado por número de tramas, longitud de cadena o número de símbolos en el diccionario de dicho candidato; longitud de cadena normalizada por el número de tramas, etc.
- Parámetros estadísticos: Calculados sobre la distribución de los costes de acceso léxico, para distintas longitudes de la lista de preselección. Así, se calculan medias y desviaciones de dichos costes, normalizados o no según los criterios vistos más arriba, para longitudes iguales al 0'1%, 1%, 10%, 25% y 50% del tamaño del diccionario usado.

Dado el desconocimiento que tenemos sobre el problema, nos interesa disponer del mayor número posible de parámetros, y dejar que sean las técnicas utilizadas las que nos ayuden a hacer la selección más efectiva. A lo largo de este capítulo haremos referencia al número de orden de los parámetros, de acuerdo con la asignación detallada en el Anexo A, mencionando el nombre concreto de los mismos cuando lo consideremos necesario para una mejor comprensión de lo descrito.

4.4.5 Análisis estadístico previo (distribuciones y correlación)

Antes de abordar el estudio en profundidad de los distintos métodos aplicados, hicimos una exploración inicial para determinar la existencia o no de correlaciones evidentes entre los parámetros

1. Realmente la primera pregunta es si el problema es o no resoluble. Lamentablemente en muchos casos, entre ellos el que nos ocupa, esta cuestión es algo desconocido a priori.
2. Con una palabra más larga, se tiene más evidencia acústica con la que comparar, es decir más información en el proceso.

disponibles y el parámetro a estimar: la posición en la que se reconoció correctamente la palabra, que pretendemos utilizar como longitud de la lista de preselección.

La exploración se centró en la aplicación de métodos descriptivos: técnicas simples de correlación y regresión lineal, simple y múltiple, por ser las más cómodas de visualizar e interpretar.

El estudio de los coeficientes de correlación entre cada parámetro y la posición de la palabra reconocida mostró valores muy bajos, en general. En la Figura 4-13 se muestra dicho coeficiente (en valor absoluto) en función del número de orden del parámetro con el que se compara. El valor máximo, es decir 1, se obtiene, evidentemente, usando la *posición correcta reconocida* como parámetro (número 5: *posicOK*) y puede observarse la ausencia prácticamente total de correlación con el resto.

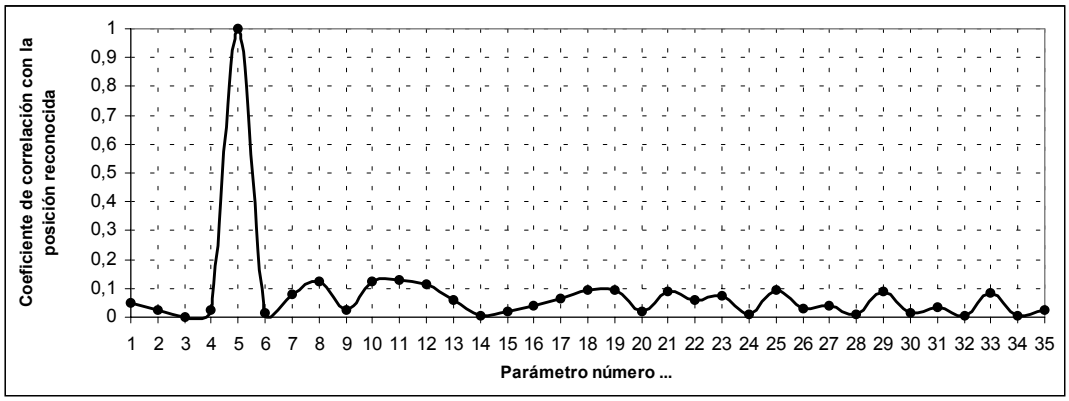

Figura 4-13: Valores del coeficiente de correlación entre los parámetros disponibles y la longitud de lista a estimar (posición correcta)

En lo que respecta a los cálculos de los coeficientes de correlación para cada par de parámetros, los resultados son los razonablemente esperados dadas las características del repertorio usado y no han producido buenos resultados.

Los análisis de regresión lineal simple y múltiple (con un número variable de parámetros de entrada) tampoco han arrojado resultados de los que se derive la utilidad directa para estimación, habiendo obtenido valores del coeficiente de determinación extremadamente pequeños, lo que invalida cualquier análisis posterior de fiabilidad. A pesar de ello, se han realizado algunos experimentos piloto de estimación de longitudes de lista a partir de estos estudios de correlación, indagando en su posible utilidad final[1]. Dichos experimentos han obtenido resultados muy pobres.

Ni siquiera el parámetro *estrella*, el que dio lugar a las primeras consideraciones sobre la viabilidad de este estudio: *el número de tramas* ha obtenido resultados notables, aunque cabe mencionar que la pendiente de la recta de regresión lineal que lo relaciona con la longitud de lista es negativo, lo que confirma, en cierto modo la idea cualitativa que teníamos de su comportamiento.

También se realizaron análisis similares para los parámetros en dominios transformados (usando normalización, escalado, etc., aplicando las ideas que se describirán en el Apartado 4.4.9.2, a partir de la página 101, al hablar de técnicas de codificación de los parámetros de entrada a la red neuronal), obteniendo resultados similares.

Los malos resultados obtenidos en los análisis realizados hace inútil cualquier investigación posterior en la línea de aplicar métodos inferenciales de análisis, por lo que abandonaremos aquí esta línea de trabajo.

4.4.6 Métodos paramétricos

En los métodos paramétricos el objetivo es imponer una relación analítica determinada entre el parámetro (o parámetros) a considerar y la longitud de la lista.

1. Básicamente se trataba de aplicar las fórmulas de correlación con los datos obtenidos sobre la base de datos de entrenamiento para estimar la longitud de lista de preselección a usar.

A la vista de los pocos resultados de utilidad práctica obtenidos con los métodos simples de estimación descritos en el apartado anterior, abordamos métodos más específicos para la tarea que nos ocupa.

Nuestro primer intento en esa dirección fue utilizar una función lineal del parámetro de control, lo cual podría parecer idéntico al análisis de regresión lineal descrito anteriormente, salvo por el hecho de que no aplicamos la metodología estándar, sino una adaptada a nuestro problema. La idea no es buscar la recta de mejor ajuste, sino la que cumpla las restricciones deseadas en nuestro problema.

Así por ejemplo, cuando usamos el número de tramas como la variable independiente, a medida que la palabra es más larga, más pequeña será la lista y viceversa. En la Figura 4-14 se muestra, por ejemplo, para todos los ficheros de la lista PEIV1000, los pares (número de tramas, posición reconocida)[1], donde aunque no se puede observar claramente una relación como la descrita, sí puede intuirse cierta tendencia[2] a disminuir la posición reconocida a medida que aumenta el número de tramas o, por lo menos, la posibilidad de utilizar listas de preselección de tamaño más pequeño para palabras más largas.

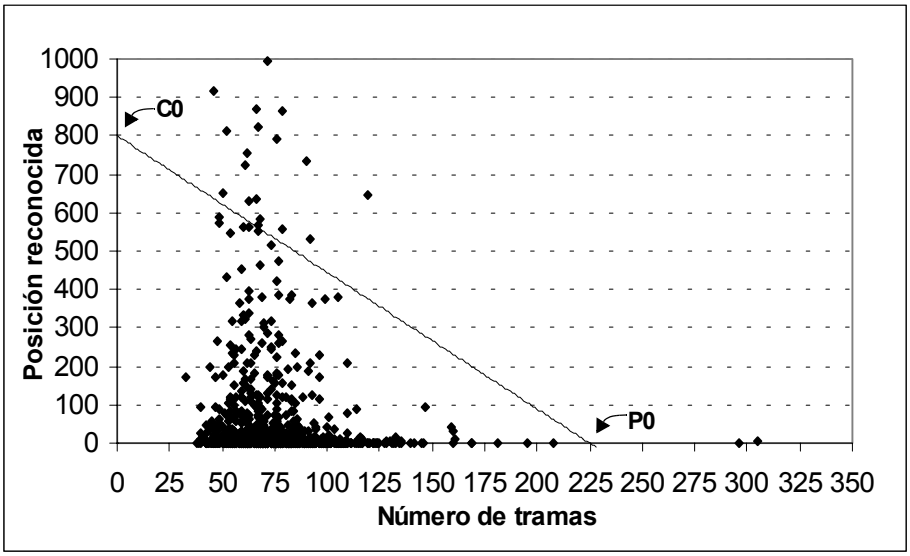

Figura 4-14: Relación entre el número de tramas y la posición en la que se reconoció cada palabra para la lista PEIV1000. Se ha superpuesto una posible recta de estimación de LongLista para C0=800 y P0=225

Así, se trataría de estimar la ecuación de la recta con la que se consiguen los resultados deseados:

$$longLista(i) = -\frac{C0}{P0} \cdot numTramas(i) + C0$$

donde *C0* y *P0* son los parámetros a calcular que, gráficamente son los valores de intersección de la recta con los ejes de coordenadas).

Evidentemente la relación analítica mostrada es extremadamente pobre, en cuanto a simple, y se haría necesaria una exploración más exhaustiva de alternativas, haciendo intervenir un mayor número de parámetros, por ejemplo, o utilizando funciones más sofisticadas de estimación. Sin embargo, comenzamos por este estudio como prospección de las posibilidades de este campo.

Únicamente se hicieron pruebas realizando la estimación de los valores de *C0* y *P0* sobre la misma lista sobre la que se evaluó su impacto, con lo que los resultados obtenidos son de auto-

1. El eje de ordenadas ha sido limitado a 1000 candidatos para facilitar la visualización.
2. En la figura no es posible apreciar la distribución frecuencial de las palabras en función de los pares (número de tramas, posición reconocida). Dicha información es fundamental por su influencia en la estrategia de estimación que perseguimos en este apartado.

comprobación (*autotest*) y sólo nos valen para tener una estimación del funcionamiento óptimo del sistema.

Para evaluar el método, se varían los valores C0 y P0 en un rango razonable (de acuerdo con los rangos de variación de los parámetros implicados) y se aplica la ecuación vista anteriormente para determinar la longitud de lista para cada palabra, calculando la tasa de inclusión obtenida y el esfuerzo medio correspondiente. Así, se admiten como *válidos* aquellos pares *(P0, C0)* en los que el error de inclusión es menor del 2%[1] y el esfuerzo medio inferior al fijo necesario para conseguir la misma tasa (que en la prueba piloto era de 216 candidatos).

El experimento exploratorio obtuvo los resultados que se muestran en la Figura 4-15. En la parte superior los ejes representan los valores de *C0* (ordenadas) y *P0* (abscisas). Para cada par de valores *(C0, P0)* se ha marcado un punto gris si se considera válido. En la parte inferior, los ejes representan los valores de *P0* (abscisas) y esfuerzo medio obtenido (ordenadas), habiéndose dibujado el punto correspondiente en gris para cada uno de los puntos marcados en la gráfica superior. El hecho de que parezcan zonas sombreadas se debe simplemente a la superposición de puntos contiguos.

A partir de esas dos gráficas de la Figura 4-15, puede verse (puntos resaltados en negro) que para aproximadamente *P0=125*, conseguimos un valor mínimo de esfuerzo medio de alrededor de 160, lo que supone una reducción de aproximadamente el 26% respecto a los 216 candidatos que usábamos en listas fijas. Este valor corresponde a un *C0=410*.

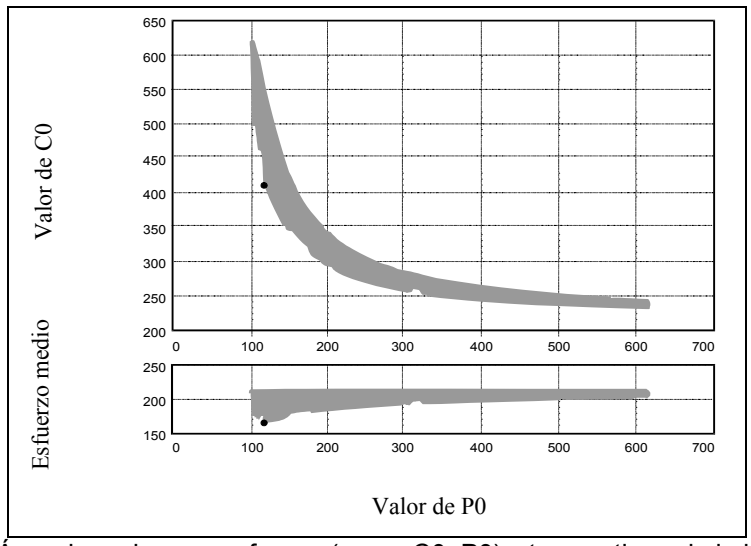

Figura 4-15: Superior: Área de mejora en esfuerzo (pares C0, P0) y tasa estimando la longitud de lista con una ecuación lineal dependiente del número de tramas.
Inferior: Área correspondiente a los esfuerzos medios obtenidos para cada uno de los pares de la gráfica superior.

Por supuesto, el principal problema en este método es la simplicidad de la función de estimación y la crítica fundamental a la prueba realizada es el hecho de que los resultados mostrados se refieren a experimentos con una única base de datos (*autotest*) que además es muy limitada.

En una implementación real habría que incluir una estimación de umbrales adicionales que añadiríamos a la estimación del sistema para mejorar la robustez al enfrentarse con ejemplos desconocidos.

Queda para trabajos futuros el desarrollar más esta línea de estimación de longitudes de lista.

1. Criterio de diseño del módulo de preselección.

4.4.7 Métodos no paramétricos

En nuestro caso, al hablar de métodos no paramétricos nos referiremos a aquellos en los que no se hace ninguna suposición acerca de la relación funcional entre las variables de entrada (en nuestro caso los parámetros disponibles descritos en el Anexo A: número de tramas, coste del módulo acústico, coste del acceso léxico, etc.) y la que pretendemos obtener (la longitud óptima de la lista de preselección, que, para cada palabra, coincide con el valor de la posición en la que fue reconocida).

Dentro de estos métodos únicamente trataremos dos alternativas: estimación basada en cálculo de tablas de corte y uso de redes neuronales, que serán tratados en los siguientes apartados

4.4.8 Métodos no paramétricos basados en el cálculo de tablas de corte

Nuestro trabajo inicial en este sentido se orientó a construir una tabla de valores en la que relacionáramos el valor de un parámetro de entrada dado (o, genéricamente, un rango del mismo[1]) con una cierta longitud de lista de preselección a usar. A esto lo llamaremos, por convención y para simplificar, *tabla de corte*, porque, en cierto modo, se usa para *cortar* la longitud de la lista a usar, al estilo de lo que haría un método de poda (*pruning*) tradicional.

La construcción de la tabla de corte sigue un proceso iterativo en el que se busca, para cada posición de la tabla de corte, aquella palabra a reconocer en la que el sistema se comporta peor (esto es, aquellas para las que la lista de preselección será mayor) de entre todas las que comparten la misma posición *i* en dicha tabla (por ejemplo, todas las que tuvieran *i=27* tramas de longitud), descartándola si es posible[2], antes de proceder a una nueva estimación de la tabla de corte. Con *descarte* nos referimos a que eliminaremos dicha palabra de la lista de ellas a considerar en el cálculo de la *tablaCorte(i)* correspondiente, con el objetivo de reducir progresivamente los valores encontrados en dicha tabla.

La tabla resultante nos daría finalmente valores concretos de longitudes de lista de preselección a utilizar en función del valor (o rango de valores) del parámetro de entrada dado (o parámetros, si la estimación se hiciera con múltiples variables).

Al igual que en el Apartado 4.4.6 donde se describía un experimento inicial para mostrar la viabilidad de los métodos paramétricos simples, hicimos lo mismo en este caso y en las mismas condiciones de prueba (*autotest*). De todos los parámetros disponibles, sólo se utilizó el número de tramas como parámetro de control, obteniendo los resultados mostrados en la Figura 4-16 en la que aparece la evolución del esfuerzo medio como una función del número de iteración del proceso, en un experimento en el que, recordamos, la lista de preselección necesaria para obtener un 2% de tasa de error era de 216 candidatos, usando el número de tramas como la variable de control.

La gráfica se ha centrado en la zona de interés y de ahí que comience a dibujarse para un valor de esfuerzo medio ligeramente superior a 216 candidatos de esfuerzo medio (obviamente, para las primeras iteraciones obtenemos tasas superiores al 98% y esfuerzos superiores al 216). La región sombreada muestra la zona para la que el esfuerzo medio calculado era inferior a 216 (que es el objetivo a batir) y la tasa de error medida inferior al 2%. Observando el gráfico, puede verse cómo, en el caso óptimo, podríamos llegar a conseguir un esfuerzo medio de alrededor de 75, es decir una reducción relativa en esfuerzo del 65%, manteniendo la tasa objetivo.

---

1. La distinción entre valor estricto del parámetro y rango del mismo es fundamental. Así por ejemplo, para parámetros escalares, como el número de tramas, el índice que recorre dicha tabla puede ser directamente el mismo valor de número de tramas. Sin embargo, en parámetros que son números reales, como por ejemplo el coste del módulo acústico, es necesario definir una división del espacio de valores que pueda tener, en rangos determinados. En este último caso, será el número de orden del rango considerado el índice con el que recorramos la tabla.
2. En este punto podemos aplicar varios criterios válidos para restringir si descartamos o no una palabra, pero en nuestro caso nos hemos limitado a permitir el descarte si el hecho de recortar la lista a ese valor sólo afecte a una palabra, ya que queremos limitar el impacto de dicha eliminación en la tasa del sistema

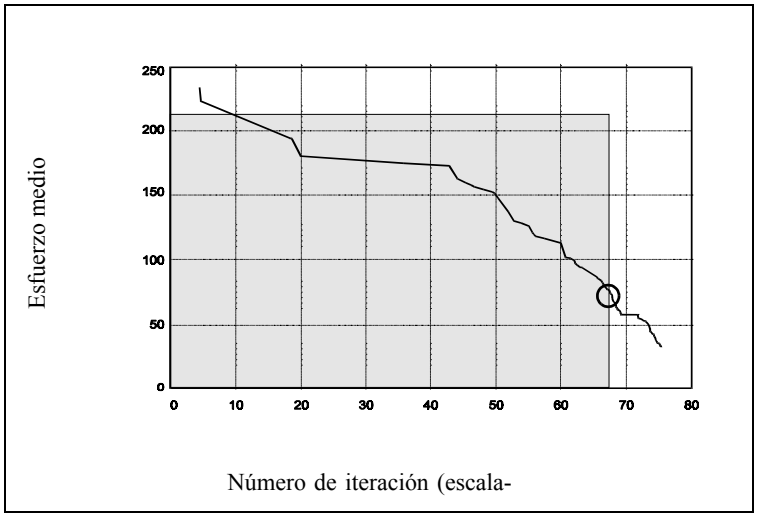

Figura 4-16: Reducción media de esfuerzo usando el número de tramas como parámetro de control en el método de construcción de tablas de corte

De nuevo este enfoque parece prometedor, pero tiene dos inconvenientes fundamentales.

- El primero es que las tablas de corte son extremadamente dependientes de los datos de entrenamiento, de forma que deberían ser suavizados a posteriori para abordar datos desconocidos, lo que seguro incrementará el esfuerzo necesario que será siempre superior a ese óptimo obtenido.
- El segundo se refiere a la *granularidad* usada para discretizar el rango continuo de algunos de los parámetros: a medida que hacemos los intervalos más pequeños, obtendremos tablas demasiado ajustadas a los datos de entrenamiento. Si elegimos intervalos mayores, habrá más ocurrencias que se verán afectadas por cambios en el umbral de corte, impuesto por la que peor resultado haya tenido, de forma que la reducción en esfuerzo medio será menor.

Además, las consideraciones hechas anteriormente respecto a que se trata de un experimento de auto comprobación (*autotest*) impiden generalizar los resultados obtenidos, y como ocurría en el caso anterior (métodos paramétricos), no abundaremos más en esta línea, quedando planteado para trabajos futuros.

4.4.9 Métodos basados en redes neuronales

Las redes neuronales son uno de los métodos más prometedores en lo que se refiere a sistemas de reconocimiento de patrones [Bishop95][Ripley96].

En nuestro caso, optamos por evaluar esta técnica, dada la naturaleza del problema de estimación que nos ocupa, es decir, dada la ausencia de conocimiento explícito acerca de la naturaleza de la relación (si es que existe) entre los parámetros de los que disponemos y la longitud que queremos estimar, y dada igualmente la existencia de datos suficientes (en principio) para entrenar una red neuronal de tamaño razonable.

En lo que sigue, discutiremos sobre algunos de los factores fundamentales a determinar en el momento del diseño de cualquier sistema basado en redes neuronales, centrándonos por supuesto en la tarea que nos ocupa. Así, analizaremos los parámetros de entrada disponibles, las distintas técnicas de codificación que podemos aplicar a aquellos, las consideraciones en el diseño de la topología y las técnicas de entrenamiento, para abordar a continuación los detalles de los experimentos realizados y las conclusiones extraídas de todos ellos.

4.4.9.1 Selección parámetros de entrada y topología

Los parámetros a usar serán los mencionados en el Apartado 4.4.4. En lo que respecta la topología, fundamentalmente habrá que decidir acerca del número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexiones entre neuronas, aparte de ciertos parámetros adicionales como el tipo de función de activación a usar y la existencia o no de neuronas de umbral (*bias*, en la nomenclatura inglesa) en el cálculo.

El uso de una u otra topología puede significar la diferencia entre el éxito y el fracaso, pero desgraciadamente no hay fórmulas ni teorías conocidas que permitan diseñar una red con garantías de éxito. A lo sumo existen ciertas recomendaciones generales [Masters93] basadas en la experiencia directa pero que carecen de apoyo teórico sólido.

En nuestro caso usaremos un perceptrón multi-capa (MLP) con 3 capas (incluyendo la de entrada) y conexiones hacia adelante, es decir, con una única capa oculta, que de acuerdo con la literatura es suficiente para realizar cualquier tarea discriminativa (en un MLP que use cualquiera de las funciones típicas no lineales de activación, basta con una capa oculta para asegurar la propiedad de *aproximación universal*).

Si una arquitectura como esa no es capaz de solucionar el problema planteado, dicho defecto no es achacable a la red en sí, sino a otros factores, como insuficientes datos de entrenamiento, insuficiente número de neuronas en la capa oculta, poca adecuación de los parámetros dados (que no son relevantes para el problema) o que la función es simplemente no determinista o no existe.

En cuanto a la decisión con respecto al número de neuronas, sólo contamos con las recomendaciones simples de las que hablamos anteriormente. En general depende de forma crítica del número de datos de entrenamiento, de la cantidad de ruido, y, por supuesto, de la complejidad de la tarea de discriminación/clasificación a la que nos enfrentemos.

Una consideración adicional es la que se refiere a la cantidad de datos disponibles para entrenamiento. Es imprescindible asegurar que podremos entrenar los parámetros de forma razonable sin caer en problemas de sobre-entrenamiento (memorización de ejemplos) o de sub-entrenamiento (incapacidad de modelar la función objetivo).

En nuestro caso, y a modo de regla simple (que adolece de la misma falta de soporte teórico comentado anteriormente), vamos a tratar de conseguir que el mínimo número de ejemplos disponibles supere en un orden de magnitud el número de parámetros de la red, con lo que tendremos que mantener el tamaño controlado. Las funciones de activación a utilizar en la red (las que introducen no linealidad) serán en todos los casos sigmoides y, con la estructura planteada, va a utilizarse también un término de umbral en la capa de entrada, cuya misión fundamental es desplazar el punto de decisión en la función de activación.

En la literatura se describen técnicas de reducción del número de neuronas en una red, pero no vamos a abordar ninguna de ellas, dado que, de entrada, nos vamos a quedar con topologías relativamente pequeñas, dadas nuestras limitaciones en número de datos de entrenamiento, aunque podríamos haber usado esa estrategia como técnica genérica de reducción del número de parámetros, si dispusiéramos de mayores bases de datos.

4.4.9.2 Técnicas de codificación de los parámetros de entrada

Una vez establecida la topología general a utilizar y los parámetros de los que dispondrá nuestra red, queda por solventar la decisión relativa al modo en que presentaremos dichos datos a nuestra red.

Una aproximación rápida podría ser entregar directamente los datos a la red, sin más preproceso. En general, esa estrategia raramente funciona y lo primero que nos debemos plantear es cómo podemos hacer más cómodo el trabajo de la red.

En nuestro caso, todas las variables son numéricas y las transformaciones que planteamos como alternativas se refieren a normalización y escalado, y uso o no de codificación en múltiples neuronas. El objetivo es siempre mejorar las condiciones numéricas del problema de optimización y asegurar que los valores usados en la inicialización y el criterio de convergencia son adecuados.

El proceso de codificación debe hacerse con cuidado, ya que puede implicar pérdida de información, en general. Si la información descartada es irrelevante, no hay problema, pero no tiene porqué ser así. Así, para variables a codificar en una única entrada (a lo que nos referiremos como *monoentrada* en lo que sigue), plantearemos cuatro estrategias de normalización:

1. Sin normalizar (método al que llamaremos *NO-NORM*)
2. Normalización entre máximo y mínimo (*NORM-MAXMIN*)
3. Normalización a una distribución de media 0 y varianza 1 (*NORM-STD*[1])
4. Normalización a una distribución de media 0 y varianza 1, limitando el rango final de valores a uno determinado (al que llamaremos *NORM-STD-CLIP*[2], al conocerse dicha técnica como *data clipping*, en la nomenclatura inglesa). La idea es similar a la comentada en el punto 3 anterior, en cuanto a la distribución usada, pero evita que valores muy alejados de la distribución (que posiblemente sean ruidosos) afecten al proceso de estimación[3].

Por otro lado, cabe la posibilidad de codificar cada parámetro en varias entradas (a lo que nos referiremos en lo que sigue como *multientrada*), cada una de las cuales representaría, básicamente, a un rango de valores de dicho parámetro. En esta estrategia, se analizó el efecto de tres factores:

1. La distribución de los rangos asignados a cada entrada: Que en una primera aproximación podría ser lineal, es decir, el rango completo del parámetro se divide en *n* segmentos iguales y durante el funcionamiento de la red, se activa la entrada correspondiente al rango en el que esté el parámetro de entrada (método al que llamaremos *BINLINEAL*[4]). El inconveniente fundamental de esta estrategia es que no atiende a la distribución del parámetro, en el sentido de que no contempla la mayor o menor concentración de valores en ciertos segmentos. Esto puede producir desequilibrios en el número de activaciones disponibles en el entrenamiento para cada entrada (lo que enlaza directamente con lo que comentaremos en el Apartado 4.4.9.3 acerca de la distribución del número de ejemplos de entrenamiento que activan cada una de las neuronas de salida).

   La alternativa consiste en diseñar los segmentos de codificación de forma que se iguale el número de ejemplos de los que dispondremos en entrenamiento para cada entrada (método al que llamaremos *BINNOLINEAL*[5]).

1. Procedente de **NORM**alización a la distribución e**ST**andar**D**.
2. Procedente de **NORM**alización a la distribución e**ST**andar**D** limitando (cortando) el rango a unos valores determinados (**CLIP**ping, en inglés), los que delimiten la mayor parte de los ejemplos de la distribución.
3. Algunos autores son reacios al uso de técnicas de limitación del rango de variación de los datos, ya que pueden implicar pérdida de información relevante. En nuestro caso se incluye la técnica por completitud.
4. Procedente de distribución en segmentos (**BIN**s, en inglés, que tiene varias acepciones entre las que figura ésa) distribuidos **LIN**ealmente
5. Procedente de distribución en segmentos (**BIN**s, en inglés, que tiene varias acepciones entre las que figura ésa) distribuidos **NO LIN**ealmente

Estos esquemas pueden suponer ciertos problemas a considerar cuando se procesa la base de datos de reconocimiento (evaluación), ya que la distribución de los datos no tiene por qué ser la misma[1]. En cualquier caso, mantendremos estrictamente el cálculo de todo tipo de parámetros necesarios sobre la lista de entrenamiento, exclusivamente.

2. El número de entradas a activar: Según esta idea, podremos activar únicamente la neurona correspondiente al rango al que pertenece el parámetro de entrada (método al que llamaremos *UNIACT*[2]), o bien utilizar una codificación *termométrica* (método al que llamaremos *TERMOACT*[3]), en la que activamos todas las neuronas de entrada desde la primera hasta la correspondiente al rango al que pertenece el parámetro de entrada.
3. El valor asignado a la/s neurona/s activada/s: La opción más común es utilizar el máximo valor normalizado a la entrada para indicar activación y el mínimo para indicar lo contrario (método al que llamaremos *CODMINMAX*[4]), aunque podríamos hilar más fino y codificar el valor correspondiente como un número real proporcional a su posición dentro del rango al que pertenezca (método al que llamaremos *CODFLOTANTE*[5]).

4.4.9.3 Codificación de la salida de la red

El objetivo de nuestra red es ofrecer una longitud de lista determinada, en función de los parámetros de entrada. La solución más simple podría ser utilizar una única neurona de salida sobre la que se imponga, durante el entrenamiento, el valor de la longitud de la lista. Evidentemente, deberíamos aplicar un escalado para ajustarnos al rango de salida que determine la función de activación seleccionada en la última capa (típicamente entre 0 y 1). De hecho, un enfoque similar ha sido implementado con buenos resultados, tal y como se describe en el Apartado 4.5.3, a partir de la página 132, en el que se usa la activación de salida de un discriminador como estimador directo de la longitud de la lista de preselección a usar.

Por otro lado, siguiendo un razonamiento similar al hecho al hablar de la codificación de los parámetros de entrada, podemos decidir codificar el valor de salida (longitud de lista) en múltiples neuronas de la última capa.

En el primer caso (una única neurona de salida), la codificación es trivial. En el segundo (múltiples neuronas de salida), cabría aplicar cualquiera de los mecanismos descritos anteriormente. Sin embargo, la situación es ahora más crítica, por lo que haremos algunas reflexiones al respecto. En la discusión que sigue supondremos que tenemos once neuronas de salida[6].

Si pensamos en codificación *BINLINEAL*, cada longitud se asignaría utilizando una distribución uniforme. Si la tarea de trabajo usa un diccionario de 10000 palabras, cada neurona de salida implicaría 10000/10=1000 candidatos adicionales a añadir a la lista de preselección. Con esta distribución lineal, y debido a la forma de la curva de la tasa de preselección, el número de ejemplos disponibles en el conjunto de entrenamiento para activar cada neurona de salida no es homogéneo. A modo de muestra concreta, en la Tabla 4-5 se incluyen el número de ejemplos de la lista PEIV1000 que activan cada una de las 11 neuronas de salida para el experimento que describiremos en el Apartado 4.4.9.6[7], donde se puede apreciar la fuerte descompensación existente: la mayor parte de las palabras se

1. Pensemos por ejemplo un caso en el que el rango de valores del número de tramas en la lista de entrenamiento varía entre 20 y 200, y en la lista de reconocimiento entre 19 y 300. Evidentemente hay que aplicar un corte en ese caso lo que puede dar problemas al dar lugar a una estimación incorrecta.
2. Procedente de **ÚNI**camente **ACT**ivar una salida
3. Procedente de usar **ACT**ivaciones con la técnica de codificación **TERM**ométrica
4. Procedente de **COD**ificación usando un **MÁX**imo y un **MÍN**imo
5. Procedente de **COD**ificación usando un valor **FLOTANTE** en el rango dado
6. Usamos las 10 primeras salidas para caracterizar 10 intervalos de longitud de lista entre 1 y 1000 candidatos, y la salida número 11 se asigna a los casos de las palabras reconocidas por encima de la posición 1000.

reconocen en las primeras posiciones de la curva de tasa de inclusión, con lo que casi todas ellas producirán durante el entrenamiento la activación de la primera neurona de salida.

Tabla 4-5: Número de ejemplos que activan cada salida con distribución lineal para PEIV1000 en las condiciones del Apartado 4.4.9.6

Neurona de salida	*Número de ejemplos*	*Neurona de salida*	*Número de ejemplos*
1	1313	7	1
2	42	8	3
3	22	9	2
4	13	10	3
5	8	11	21
6	4		

Así, y con el objetivo de ecualizar dicho número de activaciones, dado el conjunto de ejemplos disponibles, utilizaremos un mecanismo similar al *BINNOLINEAL* descrito anteriormente, con la diferencia de que, en este caso, la distribución de la longitud de los segmentos comentados, *se entrena*, de forma que sea consistente con la curva de tasa de inclusión, buscando el mismo número de ejemplos para los que se activa cada neurona de salida en entrenamiento.

En la Tabla 4-6 se muestran los límites superiores de los segmentos asignados a cada salida en el experimento del Apartado 4.4.9.6 al que nos referíamos anteriormente. Con esos datos, la primera neurona de salida se debería activar si la palabra fuera reconocida exactamente en la primera posición; la sexta se activaría para palabras reconocidas entre las posiciones 11 y 18, y así sucesivamente, hasta la última, que se activaría para palabras reconocidas a partir de la posición 449.

Tabla 4-6: Límites de la longitud de lista de preselección en función de la neurona de salida, calculadas sobre PEIV1000-TR (entre paréntesis se muestra el número de ejemplos de entrenamiento que activan cada neurona de salida)

Neurona de salida	*Límite superior del segmento*[1]	*Neurona de salida*	*Límite superior del segmento*
1	1 (607)	7	32 (60)
2	2 (154)	8	69 (79)
3	3 (59)	9	156 (70)
4	6 (117)	10	449 (66)
5	10 (70)	11	10000[2] (55)
6	18 (95)		

1. Al que más adelante nos referiremos como $longSegmento^{nolineal}(k)$
2. En este experimento preliminar, asignamos a las palabras reconocidas por encima de la posición 1000 dicho valor, pero en la salida de la red consideramos el tamaño máximo de acuerdo con el diccionario usado

Incluso con esta distribución no homogénea, el número de activaciones para cada neurona de salida no tiene por qué estar equilibrado, como indican las cifras entre paréntesis de la Tabla 4-6, pero es la mejor solución con las restricciones que tenemos.

7. En este caso cada uno de los segmentos asociados a neuronas de salida tienen una longitud de 100 candidatos.

4.4.9.4 Post-procesado de la salida de la red

En la fase de reconocimiento, es decir, en la que decidimos finalmente la longitud de la lista de preselección a usar, tenemos varias alternativas entre las que optar. La primera y más obvia es utilizar directamente la salida ofrecida por la red.

Sin embargo, es de prever que dicha salida, cuando se enfrente a una lista desconocida, genere resultados por debajo de lo necesario (longitudes de listas de preselección que no permiten obtener las tasas requeridas), de forma que es razonable plantearse aumentar de alguna manera la longitud propuesta, para incrementar la robustez del sistema.

Las alternativas que diseñamos para la estimación de la longitud *base* de la lista de preselección (que se incrementará posteriormente como veremos más adelante) se basan en dos métodos principales:

1. La neurona de salida más activada (ganadora) es la que decide (de ahora en adelante nos referiremos a este método como *GAN*[1]), asignando como longitud de lista la asociada a ella.
2. La longitud se calcula como una combinación lineal de activaciones normalizadas de las neuronas de salida multiplicadas por el límite superior del segmento correspondiente:

$$longLista(i) = \sum_{k=1}^{NNS} lonSegmento(k) \cdot act(k)$$

   Donde *NNS* es el número de neuronas de salida, *longSegmento* es la longitud asociada a cada una y *act* es el valor de activación obtenido. De ahora en adelante nos referiremos a este método como *SUMA*[2]. La justificación de esta fórmula radica en el hecho de las activaciones normalizadas de las neuronas de salida pueden interpretarse en algunos casos como estimadores de la probabilidad a posteriori, de forma que todas las neuronas *tienen algo que decir* acerca de la longitud de la lista de preselección. Esta fórmula sobreestima la longitud de lista requerida, teniendo en cuenta la forma en la que se ha entrenado la red. La ventaja en este caso, comparada con el uso del método *GAN*, es que esa sobreestimación está más *informada*, en el sentido de que depende de las salidas reales de la red, con lo que podemos esperar que produzca mejores resultados, al no tomar una decisión tan *tajante*.

Sobre estas dos longitudes básicas, los mecanismos de incremento de las mismas que hemos planteado, son los siguientes:

1. Adición de un umbral fijo a la longitud de lista propuesta.
2. Adición de un umbral proporcional a la longitud de lista propuesta.

Dichos umbrales son también calculados durante la fase de entrenamiento, imponiendo la obtención de una determinada tasa de error de inclusión[3]. En este caso, dicha condición tiene que ser más restrictiva que la impuesta al sistema final (es decir, buscando menor tasa de error), ya que estamos tratando de incrementar la robustez del mismo cuando se enfrente al conjunto desconocido de reconocimiento[4].

---

1. Procedente de considerar la neurona **GAN**adora
2. Procedente de **SUMA**torio de activaciones por longitudes
3. En el Apartado 4.4.9.8.3, en el que se detallan los experimentos finales con el sistema de estimación basado en redes neuronales se aprovecha esta idea de imponer distintas tasas objetivo a la hora del entrenamiento.
4. Idealmente deberían ser entrenados con un conjunto de datos para validación cruzada, pero la falta de suficientes datos disponibles nos hicieron utilizar la misma de entrenamiento de la red.

En la nomenclatura presentada más arriba, añadiremos el sufijo *-FIJO* a los experimentos que utilizan umbrales fijos y *-PROP* a los que usan proporcionales. Así por ejemplo, *GAN-FIJO* se referirá a un experimento usando el método de la neurona ganadora con un umbral añadido fijo.

4.4.9.5 Métodos de entrenamiento y parámetros de control de la red

En paralelo a la experimentación inicial descrita en el Apartado 4.4.9.6, se desarrollaron una serie de experimentos orientados a determinar el efecto de distintos parámetros de diseño y control de la red en el sistema (valores de inicialización de pesos, velocidad de aprendizaje η, número de iteraciones del entrenamiento, método de entrenamiento (*online* o *batch*)), obteniendo a partir de los mismos los valores que se usaron en el resto de la experimentación.

En cuanto al entrenamiento, indicar que sólo utilizaremos entrenamiento supervisado, basado en el método de propagación hacia atrás, y la estrategia *online*, en la que la actualización de los pesos se produce para cada ejemplo, en contraposición a la estrategia *batch* (también conocida como *offline*), en la que la actualización de los pesos se realiza únicamente una vez que se han presentado todos los ejemplos de entrenamiento[1].

4.4.9.6 Experimentos iniciales

Antes de abordar un estudio sistemático de las características de la implementación final del sistema de estimación de longitud de listas variables usando redes neuronales, abordamos una serie de experimentos preliminares que nos permitieran vislumbrar la potencialidad de dicha técnica [Macías99].

A partir de los resultados obtenidos se obtuvo conocimiento suficiente para abordar con más probabilidades de éxito la experimentación sistemática realizada que se detalla más adelante, y en la que nos centraremos en:

- Plantear tareas de discriminación menos ambiciosas, esto es, que tengan que decidir entre un número menor de alternativas que las abordadas inicialmente[2]. La primera idea sería diseñar una red discriminadora sencilla que distinguiera si una palabra ha sido reconocida en primera posición o en cualquier otra. Adelantando acontecimientos, queremos señalar que una aplicación inmediata de la red discriminadora simple es la asignación de valores de confianza a una palabra dada, como se tratará con más detalle en el Apartado 4.5.

  La extensión inmediata al discriminador simple, sería diseñar una estructura jerárquica de redes, de forma que entrenemos discriminadores aplicados sucesivamente y, posiblemente, apoyados en los resultados del discriminador anterior, de forma que aseguremos en todos los casos tareas que distingan únicamente entre dos alternativas y además estén razonablemente equilibradas en cuanto a número de ejemplos. Los experimentos en esta dirección mostraron que aún así nos encontramos con problemas de distribución de la base de datos de entrenamiento, con lo que no abundaremos en ella.

- Evaluar la potencialidad de cada parámetro y de cada método de codificación y topología en ese proceso discriminador, proceso que se trata en el Apartado 4.4.9.7, usando precisamente la idea de una red discriminadora sencilla.

En los apartados que siguen, se tratarán en detalle los experimentos relativos a estas líneas de trabajo.

---

1. Experimentos previos en el Grupo y algunos preliminares realizados en este trabajo de tesis desaconsejan la utilización del método *batch*.
2. Los experimentos preliminares usaban una red con 10 salidas, lo que complica de forma importante la decisión acerca de la potencia del método y la estimación de la capacidad discriminativa de los parámetros usados.

4.4.9.7 Experimentos de discriminación primera posición vs. resto

Recordando el planteamiento hecho anteriormente, la tarea inicial consiste en diseñar una red que decida si la palabra reconocida lo ha sido en primera posición (lo que para el conjunto de entrenamiento sucede el 46,95 % de las veces, en las condiciones descritas en el Apartado 4.4.9.7.1), o en cualquier posición a partir de ella (lo que sucede el 53,05% restante). Con esto obtenemos un número mucho más homogéneo de ejemplos que entrenan cada activación de salida[1].

El criterio de evaluación utilizado en estos experimentos será la tasa de discriminación obtenida, entendida como el porcentaje de aciertos de clasificación[2], para decidir acerca de la potencia del modelado de cada uno de los parámetros y de las codificaciones aplicadas.

4.4.9.7.1 Base de datos y experimento de referencia

La base de datos de los experimentos que describiremos en este apartado es VESTEL, tal y como se describe en el Apartado B.2.2, usando PRNOK5TR como conjunto de entrenamiento y PERFDV y PEIV1000 como conjuntos de reconocimiento.

El experimento de referencia utiliza modelado semicontinuo independiente del contexto con el alfabeto `alf23`[3] y el diccionario de 10000 palabras[4] en todos los casos.

4.4.9.7.2 Selección de topologías, parámetros y codificaciones

Nuestro objetivo es barrer todos los parámetros y el mayor número de combinaciones de codificación posibles, para estimar fiablemente la potencia discriminativa de cada alternativa.

La topología sólo se modificó con cambios menores, al haberse establecido una arquitectura básica con pocas variaciones (que se decidieron a partir de la realización de unos experimentos preliminares):

- Una capa de entrada con tantas neuronas de como fueran necesarias, 1 por parámetro para codificación *monoentrada* y 5, 10 o 20 si se usaba *multientrada*.
- Una única capa oculta con un número variable de neuronas: 5, 10 o 20 en todos los casos.
- Una capa de salida compuesta por una única neurona, en la que una baja activación indicaría palabra reconocida en primera posición y una activación alta, palabra reconocida en segunda o superior posición. En el entrenamiento dichos valores se fijaron a 0.1 y 0.9, respectivamente. En los experimentos iniciales usamos un umbral de discriminación situado en la mitad de esta banda, es decir, de 0,5; aunque también se evaluaron métodos estadísticos de estimación de un umbral óptimo, como se describe en el Apartado 4.4.9.8 a partir de la página 115.

Inicialmente lanzamos experimentos para cada uno de los 33 parámetros disponibles descritos en la Tabla A-1 a partir de la página 185, modificando la configuración de acuerdo a las alternativas de normalización y codificación descritas anteriormente en el Apartado 4.4.9.2, lo que hace un total de casi 3000 experimentos.

---

1. Es importante hacer notar aquí que estamos hablando de condiciones determinadas por el carácter de preselección de los módulos analizados.
2. No entraremos aquí en la discusión acerca del porcentaje de aciertos de clasificación de cada clase, aunque evidentemente habrá que atender a ellos en su momento. Ya adelantamos que los resultados obtenidos muestran valores muy equilibrados para ambas tasas.
3. Que recordamos utiliza 25 unidades alofónicas seleccionadas manualmente (23 alófonos+2 unidades de ruido), tal y como se describe en el Anexo D.2.3 a partir de la página 208.
4. 10000DV para las listas PRNOK5TR y PERFDV y 10000IV para PEIV1000, como se describe en el Anexo B.2.3 a partir de la página 190.

La decisión de hacer este barrido con tal número de alternativas surge del desconocimiento a priori de un método establecido de diseño de las redes a usar. Así, optamos por plantearlas todas, dando valores razonables a las topologías, asegurando un entrenamiento adecuado de los pesos de la red. A partir de los informes generados por un entorno automático de evaluación desarrollado en esta tesis, nuestra labor se centró en la selección de los parámetros más adecuados para la tarea de discriminación, del que ofrecemos detalles en los siguientes apartados.

4.4.9.7.3 Procedimiento de evaluación de potencia discriminativa: parámetros, topologías y codificaciones

A la hora de evaluar el impacto de cada alternativa en el rendimiento del sistema y llegar a un repertorio de parámetros adecuado, se optó por sistematizar el proceso usando la siguiente metodología:

- Se hicieron experimentos con una única entrada, para decidir acerca de la potencia discriminadora de cada parámetro y de las normalizaciones aplicables, descritas en el Apartado 4.4.9.2, a partir de la página página 102 (recordamos que nos referiremos a esta opción como *monoentrada*, en general)
- Se hicieron experimentos con varias neuronas de entrada, con el mismo objetivo que el punto anterior, tanto para distribuciones lineales como para no lineales en la codificación aplicada, como se describe en el Apartado 4.4.9.2, a partir de la página página 102 (recordamos que nos referiremos a esta opción como *multientrada* con codificación o distribución *lineal o no lineal*, según el caso)
- Se establecieron comparaciones primero entre experimentos monoentrada entre sí, multientrada entre sí (tanto para lineal como no lineal), y finalmente entre todos ellos, buscando obtener una estimación objetiva de la adecuación de cada parámetro a la tarea

Para efectuar la comparación, se optó por una metodología simplista, como se verá a continuación, pero que ha demostrado dar resultados más que notables. Dentro de cada comparación, la evaluación se hizo como sigue:

- Primero se calcula el número de veces que cada una de las alternativas superaba al resto en tasa de discriminación, tomando como muestra el conjunto de todos los experimentos realizados, parámetro a parámetro, de cara a decidir el orden de *bondad* de cada alternativa, cuantitativamente.
- Una vez decidido dicho orden se evalúa la mejora relativa que implica en cuanto a tasa de discriminación, para evaluar hasta qué punto merece la pena una u otra alternativa
- Finalmente se integra toda esta información para decidir, en cada caso, la lista definitiva de parámetros más adecuados, junto con la parametrización más ventajosa de cada uno de ellos y una medida de la relación mejora de tasa-demanda computacional

Antes de entrar en los resultados experimentales en sí, queremos destacar que los resultados de discriminación obtenidos han superado las expectativas iniciales que teníamos. Como puede verse en el ejemplo de la Figura 4-17, en la que se muestra la tasa de discriminación obtenida, para el caso *monoentrada*, de los 33 parámetros y las múltiples variantes de topología y codificación, prácticamente el 40% de ellas obtienen tasas de discriminación superiores al 50%, valor que hemos considerado como el umbral que diferencia unos resultados de discriminación fruto del azar de los que realmente se deben a que el parámetro considerado contiene información relevante para la tarea.

Igualmente las tasas máximas alcanzables superan el 70%, en todas las tareas en algún caso. Resultados similares han sido obtenidos para las otras dos estrategias fundamentales: codificación multientrada con distribución lineal y no lineal.

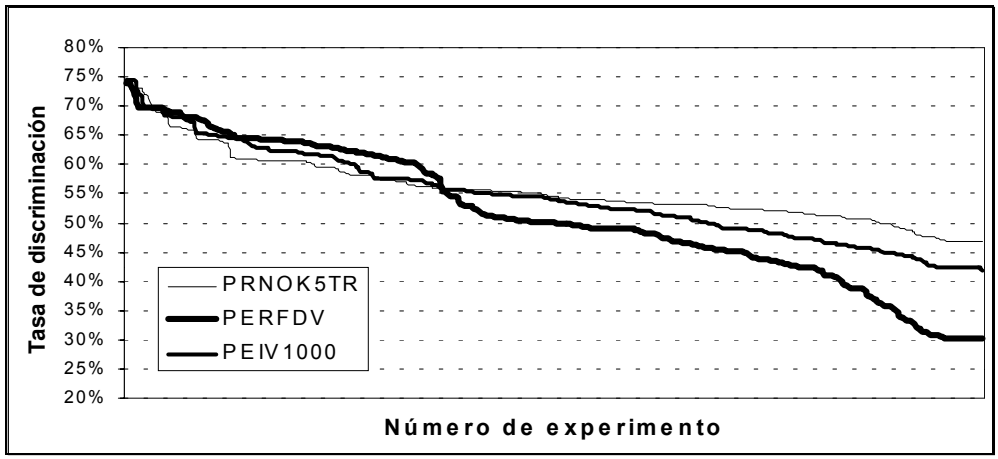

Figura 4-17: Resultados de discriminación de todos los experimentos monoentrada ordenados de mayor a menor tasa de discriminación

4.4.9.7.4 Resultados de discriminación usando un único parámetro con codificación monoentrada

En la Tabla 4-7, y siguiendo el método de decisión descrito anteriormente, se muestra el número de veces que cada combinación de topología-codificación superaba al resto (para cada conjunto de 33 parámetros posibles). Se incluyen los resultados para las tres bases de datos, a modo de referencia, aunque en nuestras decisiones sólo atenderemos a los resultados sobre la de entrenamiento, lógicamente.

Tabla 4-7: Número de veces que cada combinación topología-tipo_de_normalización superaba al resto, para el caso de topología con una única neurona de entrada. Datos para las tres listas

Normalización	*PRNOK5TR*	*PERFDV*	*PEIV1000*	*TOTAL*
NO-NORM	8	24	14	46
NORM-MAXMIN	14	22	21	57
NORM-STD	**49**	30	38	117
NORM-STD-CLIP	28	23	26	77
	99	99	99	297

Lo importante para nuestros objetivos es que, como puede observarse, tenemos una alternativa claramente ganadora: la normalización *NORM-STD* que es la que mejor comportamiento obtiene (tiene un resultado superior al resto en 49 de los 99 experimentos, casi la mitad del total).

Sin embargo el análisis comparativo de las variaciones relativas de tasa de discriminación entre los distintos casos evaluados para cada parámetro han mostrado diferencias muy poco significativas (inferiores al 1% la mayor parte de las veces). A modo de ejemplo, en la Tabla 4-8 se muestran las diferencias relativas porcentuales en tasa de error, observadas para el parámetro que mejor comportamiento obtuvo, indicando las distintas alternativas usadas (neuronas en la capa oculta y tipo de normalización). Dicho parámetro resultó ser el número 17[1]: la desviación de costes de acceso léxico para un tamaño de lista del 0.1% de la longitud del diccionario utilizado (100, en este caso, ya que el diccionario consta de 10000 palabras).

1. Remitimos al lector al Anexo A, a partir de la página página 185 donde encontrará la lista completa de parámetros usados y su significado

La visible insensibilidad del sistema a variaciones en la codificación se entiende argumentando que el factor fundamental en el funcionamiento de la red lo constituye la calidad discriminativa de los parámetros en sí, que, si son razonablemente codificados, van a presentar comportamientos similares. En este caso, podemos considerar que todas las codificaciones son *razonablemente similares*, con lo que nos quedaremos con la más simple de entre las que usan algún tipo de codificación: NORM-STD con 5 neuronas en la capa intermedia.

Tabla 4-8: Comparación de tasas de discriminación en la lista de entrenamiento para el parámetro mejor clasificado en codificación monoentrada (parámetro número 17)

Neuronas capa oculta	*Tipo de normalización*	*Tasa*	*Diferencia relativa con el menor error*
5	NORM-STD	73,56%	0,00%
10	NORM-STD	73,53%	0,13%
5	NORM-STD-CLIP	73,46%	0,39%
20	NORM-STD	73,39%	0,65%
10	NORM-STD-CLIP	73,37%	0,72%
20	NORM-STD-CLIP	73,32%	0,91%
20	NO-NORM	73,18%	1,43%
5	NO-NORM	73,13%	1,63%
10	NO-NORM	73,13%	1,63%
20	NORM-MAXMIN	72,10%	5,53%
10	NORM-MAXMIN	71,98%	5,99%
5	NORM-MAXMIN	71,77%	6,77%

La tabla específica de resultados de tasa de discriminación para los mejores parámetros en todos los experimentos realizados sobre la lista PRNOK5TR (cuyas primeras posiciones las ocupa el parámetro número 17, con las tasas indicadas en la Tabla 4-8) se incluye en la Tabla 4-9, en la que se indica el parámetro dado, la tasa de discriminacion obtenida, la topología (INTER seguido del número de neuronas de la capa oculta usadas) y codificación usada y la diferencia relativa en error de discriminación entre cada resultado y el mejor de todos. Los parámetros incluidos son, insistimos, los mejor clasificados, pero algunos aparecen duplicados (hay dos entradas para los parámetros 19, 21, 11 y 12). El motivo es nuestra intención de mostrar la mejor tasa obtenida con cada parámetro, sea cual sea la topología, y, si la topología ganadora no es la que hemos decidido utilizar (NORM-STD y 5 neuronas en la capa oculta), la tasa para ésta, de modo que se pueda ver efectivamente que las diferencias entre tasas obtenidas variando la codificación y la topología no son significativas.

Tabla 4-9: Resultados obtenidos en la tarea de discriminación con los mejores parámetros en la lista de entrenamiento en codificación monoentrada de entrada.

Parámetro	*Topología y codificación*	*Tasa*	*Diferencia relativa con el menor error*	*Diferencia con el mejor resultado para ese parámetro*
17	INTER5-NORM-STD	73,56%	0,00%	
19	INTER10-NORM-STD-CLIP	69,85%	14,06%	
19	INTER5-NORM-STD	69,00%	17,25%	2,80%
21	INTER10-NORM-STD-CLIP	66,40%	27,08%	
21	INTER5-NORM-STD	66,39%	27,15%	0,05%
11	INTER5-NORM-MAXMIN	64,44%	34,51%	
11	INTER5-NORM-STD	64,34%	34,90%	0,29%

Tabla 4-9: Resultados obtenidos en la tarea de discriminación con los mejores parámetros en la lista de entrenamiento en codificación monoentrada de entrada.

Parámetro	*Topología y codificación*	*Tasa*	*Diferencia relativa con el menor error*	*Diferencia con el mejor resultado para ese parámetro*
12	INTER10-NORM-STD-CLIP	61,02%	47,46%	
12	INTER5-NORM-STD	60,91%	47,85%	0,26%
25	INTER5-NORM-STD	60,95%	47,72%	

Volviendo a la lista de los mejores parámetros mostrada en la Tabla 4-9 (primeras posiciones sólo, claro), estos resultaron ser[1]: 17 (*desv01CosteAL*), 19 (*Desv01CosteALNormLL*), 21 (*desv1CosteAL*), 11 (*CosteAL1erCandNormLL*), 12 (*CosteAL1erCandNormNS1*), 25 (*Desv10CosteAL*), etc., lo que fue una constante a lo largo de toda la experimentación realizada: los parámetros relacionados con la desviación estándar de la distribución de costes de acceso léxico de las palabras colocadas en las primeras posiciones de la lista de preselección son los más relevantes en la tarea discriminativa. Este último dato es sumamente interesante, ya que confirma la experiencia previa de algunos autores que utilizan precisamente medidas relacionadas con la dispersión de costes o probabilidades en sistemas de estimación de fiabilidad de reconocimiento. En estos parámetros se verificó igualmente que su poder discriminativo disminuye según aumentamos el número de elementos sobre los que se calcula (el número de candidatos de la lista de preselección usados para estimar su valor).

4.4.9.7.5 Resultados de discriminación usando un único parámetro con codificación multientrada con distribución lineal (BINLINEAL) y no lineal (BINNOLINEAL)

La misma metodología de decisión de potencia discriminativa usada en el apartado anterior se usó para evaluar el comportamiento de los parámetros para los experimentos con codificación multientrada y distribución lineal y no lineal, para la lista de entrenamiento PRNOK5TR.

De los resultados obtenidos (no mostrados aquí por simplificar el documento) se vuelve a verificar que las variaciones en la topología no parecen afectar significativamente el funcionamiento de unas codificaciones frente a otras, aunque el uso de codificación termométrica con valores flotantes en las entradas (CODFLOTANTE+TERMOACT) produce siempre los mejores resultados.

De la misma forma, el análisis comparativo de las variaciones relativas de tasa de discriminación entre los distintos casos han mostrado diferencias muy poco significativas en todos los casos de variación de topología y mecanismo de codificación, siendo de nuevo el parámetro 17 el que mejor tasa obtuvo en función de las distintas alternativas usadas y siendo los parámetros más significativos de nuevo los mismos.

Lo más destacable a este respecto es que las diferencias con las tasas obtenidas en el caso monoentrada no son significativas, como puede observarse en la Tabla 4-10 en la que se compara el mejor caso multientrada con nuestra opción monoentrada.

Tabla 4-10: Diferencia relativa de tasa de error entre los dos mejores resultados de codificación y topología para el parámetro 17

Neuronas primera capa	*Neuronas capa oculta*	*CODIFICACIÓN*	*Tasa*	*Diferencia relativa con el menor error*
20	20	CODFLOTANTE+TERMOACT NOLIN	73,61%	0,00%

1. Remitimos al lector al Anexo A, a partir de la página página 185 donde encontrará la lista completa de parámetros usados y su significado

Tabla 4-10: Diferencia relativa de tasa de error entre los dos mejores resultados de codificación y topología para el parámetro 17

Neuronas primera capa	*Neuronas capa oculta*	*CODIFICACIÓN*	*Tasa*	*Diferencia relativa con el menor error*
1	5	NORM-STD	73,56%	0,20%

4.4.9.7.6 Resultados de discriminación sobre las listas de evaluación

En este apartado daremos una breve reseña de resultados de discriminación sobre las listas de reconocimiento: PERFDV y PEIV1000, para evaluar hasta qué punto los resultados obtenidos sobre el entrenamiento son extensibles, en cuanto a capacidad discriminativa, a datos no observados previamente.

Para tener una perspectiva razonable de la cuantía de las diferencias observadas, se incluyen los siguientes datos:

- Tasa de discriminación obtenida por el mejor de los parámetros con la mejor codificación posible observada
- Tasa de discriminación del mejor representante de cada uno de los tres enfoques de codificación de entradas vistos más arriba (en los apartados 4.4.9.7.4 y 4.4.9.7.5)
- Tasa de discriminación para el parámetro y la codificación seleccionada a partir de la experimentación sobre la lista de entrenamiento: parámetro número 17 con codificación monoentrada NORM-STD.

En las tablas 4-11 y 4-12 se incluyen los resultados para las listas PERFDV y PEIV1000, respectivamente. A la vista de los mismos, es evidente que hemos perdido bastante tasa relativa a la obtenida en la lista de entrenamiento y ni siquiera la codificación seleccionada es la mejor, pero los resultados entran dentro de lo razonable y muestran que el proceso de discriminación obtiene resultados esperables sobre bases de datos desconocidas, sobre todo si mencionamos que las listas de parámetros con mayor poder discriminador son muy parecidas a las obtenidas sobre el conjunto de entrenamiento, como se verá más adelante. Igualmente es de destacar como el caso de PEIV1000 está mucho más próximo en tasa a la obtenida en la lista de entrenamiento y podemos considerar que la codificación que vamos a usar está entre las mejores que podríamos haber elegido, obteniendo con la codificación seleccionada prácticamente la misma tasa que la mejor clasificada (aumentamos el error únicamente un 1,33%).

Tabla 4-11: Resultados de discriminación sobre PERFDV con el parámetro 17.

Neuronas primera capa	*Neuronas capa oculta*	*CODIFICACIÓN*	*Tasa*	*Diferencia relativa con el menor error*
10	5	CODFLOTANTE+CODMAXMIN NOLIN	73,22%	0,00%
10	5	CODFLOTANTE+CODMAXMIN LIN	72,18%	3,88%
1	10	NO-NORM	70,53%	10,04%
1	5	NORM-STD	65,19%	29,99%

Tabla 4-12: Resultados de discriminación sobre PEIV1000 con el parámetro 17.

Neuronas primera capa	*Neuronas capa oculta*	*CODIFICACIÓN*	*Tasa*	*Diferencia relativa con el menor error*
1	10	NORM-STD-CLIP	74,47%	0,00%
10	5	CODFLOTANTE+CODMAXMIN NOLIN	74,41%	0,24%
1	5	NORM-STD	74,13%	1,33%
20	5	CODFLOTANTE+CODFLOTANTE NOLIN	72,18%	8,97%

Para el hecho de conseguir resultados ligeramente mejores en PEIV1000 que en los de la lista de entrenamiento (tasa máxima de un 74’47% frente a un 73’56%) no hemos encontrado ninguna justificación objetiva, salvo indicar que el estudio de bandas de fiabilidad muestra solape entre ambos casos.

4.4.9.7.7 Conclusiones sobre los parámetros más discriminativos

Para comenzar, incluimos en este apartado, a modo de resumen, la lista de los parámetros con los que se obtuvieron los mejores resultados de discriminación para cada una de las bases de datos tratadas y cada una de las tres grandes alternativas (monoentrada, multientrada lineal y multientrada no lineal). Los datos sobre las listas de reconocimiento se incluyen, para mostrar que los parámetros muestran un comportamiento consistente incluso para datos desconocidos.

Indicaremos un cierto número de parámetros en cada caso, y sólo nos referiremos al número de orden asignado a cada uno en el Anexo A, ya que lo que nos interesa en primera instancia es verificar la concordancia razonable entre los resultados obtenidos.

Para monoentrada:

- PRNOK5TR: **17 19 21 11** 12 10 6 25 29 33 3 7 1 4 32 28 24
- PERFDV: **17** 13 27 32 1 **19 11 21** 7 12 18 29 22 25 10
- PEIV1000: **17 21 19 11** 12 25 29 10 18 6 33 7 22 1 32 13

Para multientrada con distribución lineal:

- PRNOK5TR: **17 19 21 11** 12 10 6 25 29 33 2 7 32 28 1 24
- PERFDV: **19 17 21 11** 10 22 13 27 23 33 29 12 25 7 18 6
- PEIV1000: **17 19 21 11** 12 10 18 23 6 25 29 7 22 8 26 27

Para multientrada con distribución no lineal:

- PRNOK5TR: **17 19 21 11** 10 6 25 12 29 33 3 7 1
- PERFDV: **17 11 19 21** 13 14 27 6 7 10 25 33 26 29
- PEIV1000: **17 19 11** 12 **21** 10 23 18 6 8 30 7

Como puede verse, los resultados muestran una similitud notable, lo que de nuevo confirma la consistencia de la capacidad discriminativa de los parámetros seleccionados.

Para terminar, recopilaremos las conclusiones generales acerca de la potencia de los parámetros utilizados, a la luz de los resultados obtenidos por los resultados de los experimentos llevados a cabo:

- Los parámetros más potentes son los relacionados con la desviación de la distribución estadística de los costes de acceso léxico de la lista de preselección (*Desv*CosteAL* y *Desv*CosteALNormLL*), bajando su potencia discriminativa a

medida que incrementamos la longitud de lista sobre la que se hace el cálculo. Esto puede explicarse entendiendo que a medida que incrementamos el número de elementos, se produce un suavizado importante de los valores obtenidos.

- Los parámetros relacionados con la longitud de la palabra tienen un comportamiento desigual: el número de tramas (*numTramas*) obtiene tasas medias cercanas al 57%, mientras que la longitud de la cadena fonética (*longLattice*) alcanza aproximadamente el 60%
- De los parámetros relacionados con la probabilidad acústica, únicamente el *costePSBU* supera la tasa del 55%, quedando los otros muy alejados
- Los parámetros relacionados con el coste del acceso léxico del primer candidato obtienen buenos resultados cuando están normalizados (*costeAL1erCandNormNT*, *CosteAL1erCandNormLL* y *CosteAL1erCandNormNS1*, con tasas entre el 61% y el 64%), aunque el coste sin normalizar (*costeAL1erCand*) también funciona razonablemente, consiguiendo algo más del 57%
- Los parámetros relacionados con las medias de los parámetros calculados a partir de los costes de acceso léxico (*media*costeAL**) funcionan muy mal en general, por lo que no formarán parte del conjunto base sobre el que experimentaremos en el siguiente apartado

4.4.9.7.8 Agrupamiento de parámetros

Nuestro siguiente objetivo es, evidentemente, intentar incrementar la potencia discriminativa del sistema combinando varios parámetros de entrada a la red.

De los mejores resultados para la lista de entrenamiento, para todas las parametrizaciones, se hizo una selección inicial de parámetros que fueron los siguientes: 17, 19, 21, 11, 12, 10, 6, 25, 29, 33, 3, 7, 1 y 4 de acuerdo con los resultados obtenidos en la experimentación previa.

Se verificó que al utilizar un repertorio amplio de parámetros de entrada, las tasas de discriminacion subían consistentemente de tasa para la lista de entrenamiento, pero bajaba de forma también consistente para las de evaluación. La explicación más plausible es que hay parámetros que confunden al discriminador, impidiendo una adecuada generalización.

Dada la tendencia a obtener peores tasas a medida que aumentábamos el número de parámetros, decidimos quedarnos con la mitad aproximadamente (8) y así contribuir a robustecer el sistema y no agravar el efecto en las listas de reconocimiento.

La metodología final de agrupación consistió en partir del parámetro mejor clasificado, Desv01CosteAL (el número 17), entrenando redes añadiendo cada vez uno más de los pertenecientes a la lista base, el que mayor mejora relativa de tasa producía en cada instante.

De este proceso se llegó a la siguiente lista definitiva: *Desv01CosteAL* (17), *CosteAL1erCandNormLL* (11), *LongLattice* (3), *CosteAl1erCand* (7), *NumTramas* (1), *NumSimb1erCand* (6), *CostePSBU* (4), *Desv10CosteAL* (25)

Puede parecer sorprendente que hayan desaparecido los parámetros 19 y 21, que eran de los más discriminativos. La explicación es que el 19 y el 21 están fuertemente correlados con el 17, con lo que no aportan mucha más información que aquél.

En la Tabla 4-13 se muestran las tasas finales de discriminación obtenidas para cada una de las listas de la base de datos VESTEL. Las mejoras obtenidas no son tan notables como sería deseable, sobre todo si atendemos a la disminución relativa de error, pero hemos conseguido incrementar los resultados de partida en todos los casos[1]. Igualmente las diferencias no son estadísticamente

significativas, pero mantendremos el repertorio de 8 parámetros por la previsible mayor robustez del proceso de decisión de la red neuronal, al contar con mayor cantidad de información.

Tabla 4-13: Tasas finales de discriminación obtenidas (entre paréntesis se incluyen los márgenes de error para una fiabilidad del 95%)

Lista	*1 Parámetro (desv01CosteAL)*	*8 Parámetros finales*	*Disminución relativa de error*
PRNOK5TR	73’56% (±1’13%)	75’30% (±1’11%)	6’58%
PERFDV	65’19% (±1’86%)	68,23% (±1’82%)	8’73%
PEIV1000	74’48% (±2’26%)	74,54% (±2’25%)	0,24%

4.4.9.8 Experimentos de estimación de longitud de lista con la red completa y los parámetros definitivos

En este apartado detallaremos los experimentos finales realizados utilizando redes neuronales para estimar la longitud de lista de preselección a usar en experimentos de reconocimiento (a diferencia de lo visto hasta ahora, que eran de discriminación). La lista definitiva de parámetros a usar está compuesta por los ocho seleccionados en el Apartado 4.4.9.7.8. La asunción en este caso es que si dicho conjunto se comporta correctamente en tareas de discriminación sencilla, también lo harán en nuestra tarea final objetivo.

La decisión de la topología usada se basó en experimentos previos, quedando finalmente 8 entradas (para los 8 parámetros con codificación NORM-STD), 5 en la capa intermedia y 10 de salida, a las que se aplicó el método de distribución no homogénea de longitudes de lista de preselección discutida en el Apartado 4.4.9.3, quedando finalmente tal y como aparece en la Tabla 4-14.

Tabla 4-14: Longitudes de lista asignadas a neuronas de salida. Experimento final

Salida #	*LongLista*	*Salida #*	*LongLista*
0	1	5	13
1	2	6	23
2	3	7	51
3	5	8	134
4	8	9	10000

4.4.9.8.1 Bases de datos y experimento de referencia

En la evaluación final se usaron todos los conjuntos de datos de VESTEL (PRNOK5TR como base de datos de entrenamiento, tanto de modelos acústicos y acceso léxico como de la red neuronal; y PERFDV y PEIV1000 como bases de datos de reconocimiento, considerándolas por separado dadas las diferencias notables entre ambas).

En todos los casos se usó modelado semicontinuo independiente del contexto con el alfabeto `alf23` y los diccionarios de 10000 palabras.

1. El distinto comportamiento para las dos bases de datos de evaluación (PERFDV y PEIV1000) se debe a las distintas condiciones que presentan cada una de ellas en cuanto a su composición (como se describe en el Anexo B.2 a partir de la página 189.

4.4.9.8.2 Metodología de evaluación

El enfoque adecuado para la evaluación de los resultados obtenidos fue uno de los aspectos más complicados de decidir y constituyen otra de las propuestas de metodologías de evaluación de esta tesis.

Si consideramos el mecanismo de decisión utilizado para estimar la longitud de lista de preselección a partir de la salida de la red, es evidente que, en cada caso, obtendremos una medida de tasa de preselección determinada y un esfuerzo medio asociado, es decir, un único punto en el espacio (*esfuerzoMedio x tasaInclusión*). Nuestro experimento de referencia con longitudes de listas de preselección fijas nos ofrece en principio una curva completa de tasas de inclusión, es decir, todos los puntos posibles del espacio (*longFija x tasaInclusión*), aunque nosotros optamos por fijar como punto de trabajo el correspondiente a una tasa de inclusión del 98%.

A la hora de comparar ambos enfoques podríamos tomar como referencia los dos puntos aislados mencionados. Si con redes neuronales obtenemos una tasa mayor y un esfuerzo medio menor que la longitud fija correspondiente al 98% de tasa de inclusión, podríamos concluir que mejoran claramente el rendimiento del sistema de longitudes fijas. Si el caso es el contrario: tasa menor y esfuerzo medio mayor, tendríamos un sistema peor. Ahora bien, si una de las cifras es mejor y la otra peor, sería difícil decidir qué sistema tiene mejor rendimiento.

Incluso en el caso de que la mejora afecte a ambas cifras, queda por plantear el problema de la *sensibilidad* de los resultados a posibles variaciones en los parámetros de la red. La comparación *puntual* no ofrece información sobre dicha sensibilidad y es difícil determinar si el dato obtenido seguirá una tendencia uniforme de mejora o empeoramiento en el mismo experimento para condiciones de la red ligeramente distintas o en experimentos distintos, con lo que es imprescindible extender el análisis de resultados de modo que obtengamos una visión más amplia de la comparación a realizar. En nuestro trabajo no haremos un estudio de la sensibilidad de forma explícita, sino que nuestra estrategia será usar los mecanismos de control del esfuerzo medio que nos proporcionan los umbrales (fijo o proporcional) usados por los métodos de post-procesado de la salida de la red (descritos en el Apartado 4.4.9.4 a partir de la página 105) para obtener información relacionada con dicha sensibilidad. En cada experimento realizado con el sistema básico se obtiene un único valor de aquellos umbrales, pero si decidiéramos variar su valor en un rango determinado, obtendremos una sucesión de puntos (*esfuerzoMedio x tasa*) que, en cierto modo, serían asimilables a la construcción de una curva de tasa de inclusión *artificial* similar a la obtenida con el uso de listas de longitud fija.

La comparación a realizar se basará pues, en la verificación de si se cumple o no que el sistema basado en redes neuronales presenta una curva superior, en todo el rango analizado, a la del otro. El espacio de comparación será, obviamente, (*esfuerzoMedio x tasa*) en el caso del sistema basado en redes y (*longFija x tasa*) en el otro, al ser el esfuerzo medio y la longitud fija unidades similares en medición de coste computacional.

En cuanto a la decisión sobre el rango de variación del parámetro entrenado, se optó por hacer una exploración local para conseguir ver el funcionamiento del sistema en la zona de interés que, como decíamos antes, está situada alrededor de tasas próximas al 98%. En concreto, hemos decidido variar el parámetro de control para obtener tasas entre un 96’5% y un 99’5%, que corresponderán a unos esfuerzos medios determinados.

Al margen de abordar el estudio descrito, es necesario igualmente hacer medidas típicas de tasa y esfuerzo medio, así como calcular las mejoras porcentuales conseguidas en ambas dimensiones, naturalmente.

4.4.9.8.3 Control de los experimentos

En los experimentos realizados, se han utilizado dos elementos de control adicional del proceso de estimación de longitudes de lista, que describimos a continuación:

- Longitud de lista asociada a la última neurona de salida: Si revisamos la descripción de los métodos de post-procesado de la salida de la red para obtener la longitud final de lista reconocida (del Apartado 4.4.9.4 a partir de la página 105) y la argumentación acerca de las asignaciones de longitudes de lista a neuronas de salida (del Apartado 4.4.9.3 a partir de la página 103), recordaremos que a cada neurona de salida de la red se le asignaba una longitud de lista de preselección determinada. En su momento no hicimos referencia detallada al caso de la última neurona de salida a la que, en principio, habría que asignar una longitud de lista igual al tamaño del diccionario utilizado (10000 en el caso que nos ocupa). Ahora bien, dicha longitud puede ser excesiva en general, ya que aún en el caso peor, todos los sistemas alcanzan el 100% de tasa de inclusión con longitudes de lista bastante inferiores al tamaño del vocabulario. Éste fue el motivo que nos llevó a plantear la posibilidad de usar distintas longitudes de lista asignadas a la última neurona de salida. Concretamente, se realizaron experimentos con longitudes 0, 2500, 5000, 7500 y 10000.
- Tasa de inclusión objetivo en el proceso de entrenamiento: En el Apartado 4.4.9.4 se comentó que para estimar los umbrales (fijo o proporcional) a usar en el post-procesado de la salida del sistema basado en redes, era razonable utilizar una tasa de inclusión objetivo mayor de la que pretendemos obtener en el sistema final. Así, otro de los elementos de control que usamos en el proceso de entrenamiento fue justamente éste, imponiendo tasas objetivo del 98%, 98’5%, 99% y 99’5%.

Además de las variaciones descritas y debido a la interrelación entre ambas, hay una adicional que podemos contemplar: En el momento de la estimación de los umbrales para la tasa objetivo de que se trate, estaremos usando una longitud determinada asociada a la última neurona. Así, puede ser razonable plantear el uso de umbrales obtenidos para una longitud de última neurona distinta a la que se use en el proceso de reconocimiento.

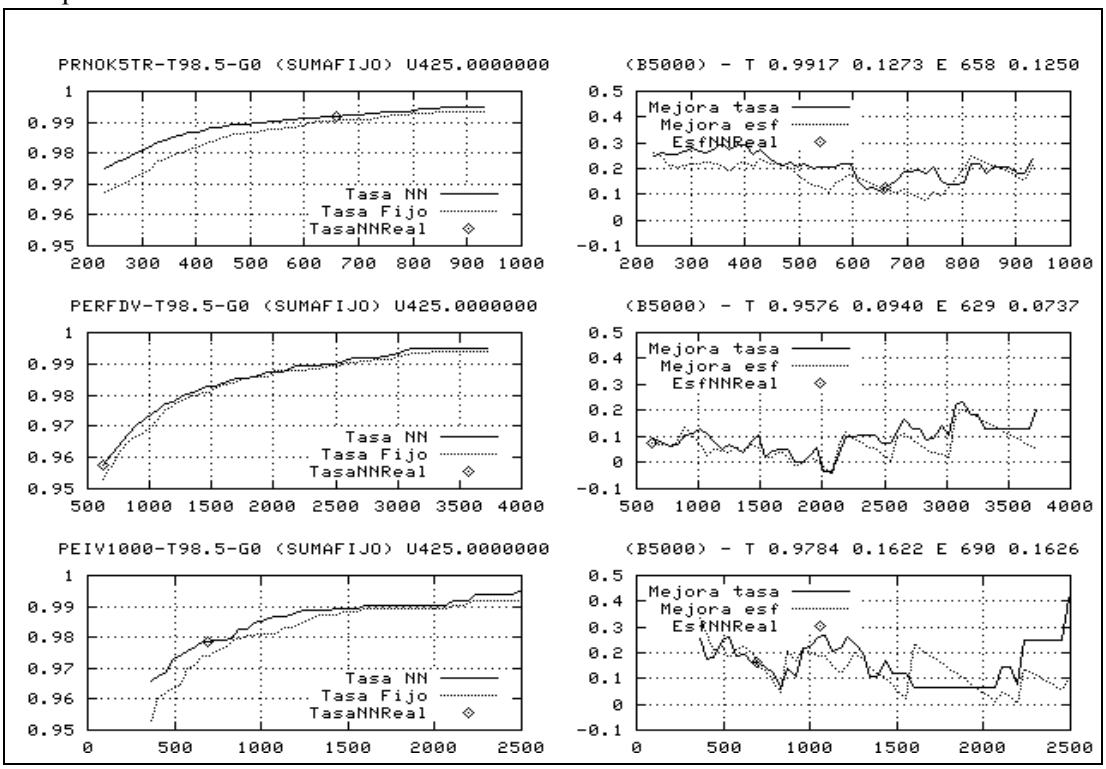

Figura 4-18: Gráfica típica para la evaluación del sistema de estimación de longitud de lista basado en redes neuronales

4.4.9.8.4 Descripción de las gráficas obtenidas y detalle del procedimiento

Para ayudar a la comprensión de los resultados obtenidos y la discusión posterior, incluimos en este apartado una descripción completa de las gráficas de evaluación que proponemos y damos los detalles completos de su proceso de construcción.

En la Figura 4-18 se muestra una gráfica típica de las que se verán más adelante. Si la interpretamos como una tabla de tres filas y dos columnas, en las que cada celda contiene un gráfica con dos curvas, la descripción de cada elemento es como sigue:

- La fila superior muestra los datos para la lista PRNOK5TR, la intermedia los de PERFDV y la inferior los de PEIV1000.
- La columna de la izquierda muestra las curvas de tasa de inclusión para el método de longitud de lista fija (línea discontinua, `Tasa Fijo`) y la curva artificial construida al variar el parámetro de control del algoritmo de cálculo de longitud de lista para el sistema basado en redes neuronales (línea continua, `Tasa NN`). El eje de abscisas muestra la longitud de lista fija usada o el esfuerzo medio calculado. El de ordenadas, la tasa de inclusión, naturalmente. Además, aparece marcado un punto fijo con la etiqueta `TasaNNReal` que muestra el punto concreto que se obtiene al aplicar el sistema basado en redes, usando los parámetro de control decididos en cada caso: la tasa objetivo y el valor de longitud de lista asignada a la última neurona en el proceso de entrenamiento y dicha longitud en el proceso de reconocimiento.
- La columna de la derecha muestra la evolución de la mejora relativa (en tanto por uno) tanto en error de inclusión (`Mejora tasa`) como en esfuerzo (`Mejora esf`) del sistema basado en redes frente al otro, en función del esfuerzo medio calculado en cada caso. Igualmente aparece un punto etiquetado como `EsfNNReal` que corresponde al punto exacto en el que se encuentra el sistema basado en redes.

Analizando ahora las etiquetas textuales de la figura, se han incluido los siguientes datos (nos referiremos a cualquiera de las tres líneas que titulan cada fila de la Figura 4-18:

- El esquema genérico de nomenclatura para cada línea es:

```
lista-Ttt.tt-Ggggg (método) Uuuuu.uuuuu (Bbbbb) - T p.pppp r.rrrr E eeee a.aaaa
```

- En primer lugar aparece el nombre de la lista dada (`lista`) que puede ser PRNOK5TR, PERFDV o PEIV1000
- A continuación, tras la `T`, la tasa objetivo (`tt.tt`) usada en el entrenamiento de los umbrales fijo o proporcional usados, en tanto por ciento
- A continuación, tras la `G`, la longitud de lista asignada a la última neurona (`gggg`) en el proceso de entrenamiento
- Seguidamente, entre paréntesis, se indica el método de post-proceso utilizado para obtener la longitud de lista final para cada palabra (cualquiera de los descritos en el Apartado 4.4.9.4)
- Después, tras la U, el valor umbral calculado (fijo o proporcional) estimado en ese experimento (`uuuu.uuuu`)
- Más adelante, entre paréntesis tras la B, el valor de longitud de lista asignado en el proceso de estimación de longitudes a la última neurona (`bbbb`) que, como discutimos más arriba no tiene por qué ser igual que el usado en entrenamiento. Este parámetro, junto con la longitud de lista y la tasa objetivo usados en el entrenamiento son los que producen el punto marcado como `TasaNNReal`

- A continuación, tras la segunda T de la línea, aparece la tasa de inclusión en tanto por uno obtenida por el sistema de redes neuronales (`p.pppp`), así como la mejora relativa de error (medida también en tanto por uno) obtenida frente al sistema de longitud fija (`r.rrrr`) que presenta el mismo esfuerzo medio. Un valor positivo indica mejora en tasa, y un valor negativo, empeoramiento.
- Finalmente, tras la E, se indica el esfuerzo medio obtenido por el sistema basado en redes neuronales (`eeee`), seguido del ahorro relativo obtenido (`a.aaaa`) al comparar con el sistema que usa listas de longitud fija que obtiene la misma tasa de inclusión. Un valor positivo indica disminución (ahorro) de esfuerzo, y un valor negativo, incremento (mayor coste computacional).

Así, leyendo la Figura 4-18, podemos decir:

- Usamos el método de post-proceso SUMAFIJO
- En el entrenamiento, obtenemos el umbral fijo buscando una tasa objetivo del 98’5% al usar una longitud asignada a la última neurona de 0 candidatos, umbral que resultó ser igual a 425
- En el proceso de reconocimiento, se usó una longitud asignada a la última neurona de 5000 candidatos
- Para la lista PRNOK5TR, se obtuvo una tasa de inclusión del 99’17% con un esfuerzo medio de 658 candidatos, lo que supone una disminución del error en un 12’73% al comparar con la tasa obtenida por el sistema de listas fijo con longitud igual al esfuerzo medio (658). La mejora en esfuerzo, es decir, la disminución relativa de coste computacional es de un 12’5%, al comparar el sistema de longitud fija con tasa igual a la obtenida con el de redes (99’17%). Además, si atendemos a la gráfica de la derecha de la primera fila, observaremos que la mejora es consistente y positiva a lo largo de todo el intervalo considerado. Nótese igualmente la posición del punto que muestra el resultado obtenido con el sistema de redes neuronales.
- Para la lista PERFDV, se obtuvo una tasa de inclusión del 95’76% con un esfuerzo medio de 629 candidatos, lo que supone una disminución del error en un 9’4%. La mejora en esfuerzo es de un 7’37%.
- Para la lista PEIV1000, se obtuvo una tasa de inclusión del 97’84% con un esfuerzo medio de 690 candidatos, lo que supone una disminución del error en un 16’26%. La mejora en esfuerzo es de un 16’26%, para un sistema de longitud fija con tasa igual a la obtenida con el de redes (97,84%).

A pesar de la aparente dificultad de análisis de las gráficas descritas y de la gran cantidad de información que contienen, su uso permite una identificación rápida de la bondad de un enfoque algorítmico dado: Mirando las gráficas de tasa, si la curva continua está por encima de la discontinua, nos encontramos con un sistema basado en redes mejor que el de longitud fija. A partir de ahí, la gráfica asociada en la parte derecha nos muestra cuantitativamente la mejora para todo el rango estudiado y, finalmente, los valores concretos del punto de trabajo en el que nos encontramos nos vienen dados en el título de dichas gráficas.

4.4.9.8.5 Resultados

Tras la realización de los experimentos pertinentes, planteando todos los enfoques algorítmicos y todas las alternativas discutidas, se observaron los siguientes efectos:

- Los sistemas basados en el método de la ganadora (GANFIJO y GANPROP) se mostraron totalmente incapaces de ofrecer buenos resultados. En la Figura 4-19 incluimos a modo ejemplo el caso del método de post-proceso GANAFIJO. Para la lista de entrenamiento conseguimos prácticamente el 98% buscado, con un esfuerzo de 619

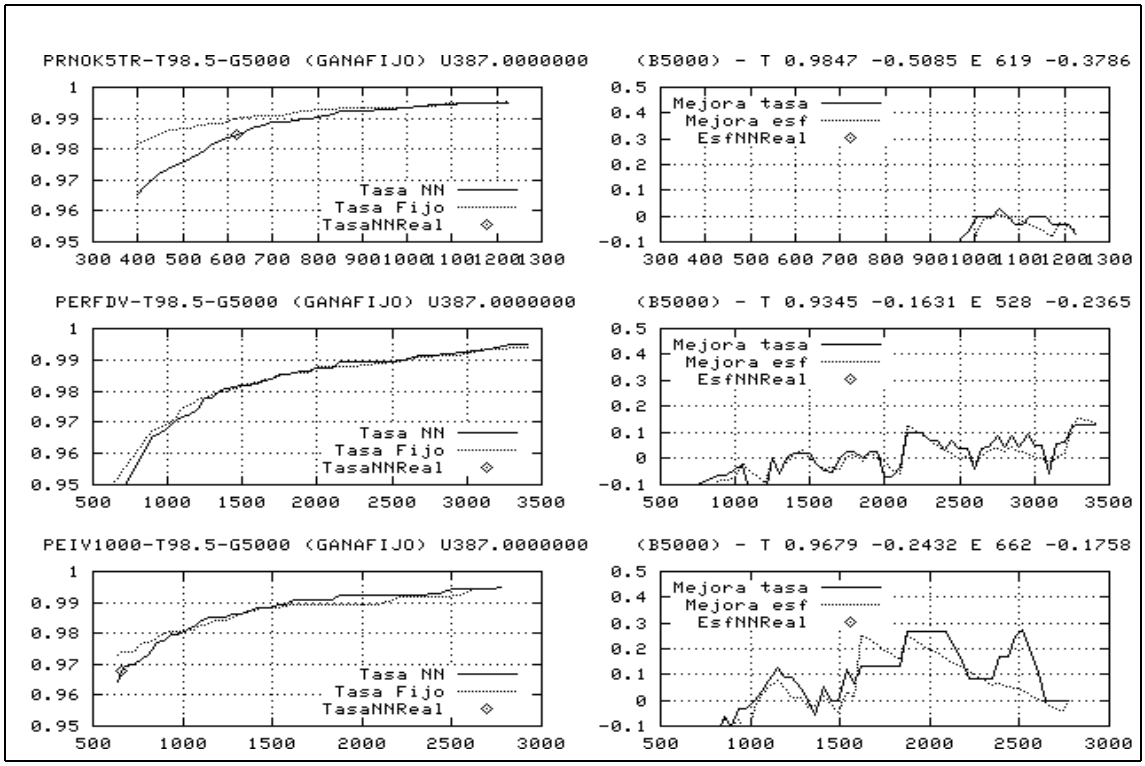

Figura 4-19: Ejemplo de resultado para método GANAFIJO con tasa objetivo en entrenamiento del 98'5% y usando 5000 candidatos para la última neurona en entrenamiento y reconocimiento

candidatos, pero en ese punto tenemos un empeoramiento en error de casi un 51% y de un 38% en esfuerzo. La misma consideración cabe hacer para las otras dos listas. Mucho más acusado es este efecto para el método GANAPROP (que funciona en general peor que GANAFIJO).

El motivo de este mal comportamiento es la falta de precisión del proceso de discriminación cuando lo extendemos a 10 clases (cada una de las 10 salidas) produciendo una degradación notable de las tasas de acierto obtenidas. Los métodos basados en la neurona ganadora dependen fuertemente de dicha capacidad discriminadora, de modo que si ésta es baja, conseguiremos resultados muy pobres.

- Los sistemas basados en el método del sumatorio presentan resultados mucho mejores y en muchos casos superan con claridad al método de listas de longitud fija. Lo más importante de los experimentos realizados es la consistencia de los resultados, es decir, el comportamiento es razonablemente homogéneo en todos los casos y responde a los mismos patrones.
- Los experimentos que usan el método SUMAFIJO proporcionan buenos resultados para las tres listas (ejemplos de los mismos pueden verse en la Figura 4-20), mientras que SUMAPROP no los consigue para PERFDV, de modo que centraremos nuestro interés en SUMAFIJO.

Del estudio detallado de todos los experimentos basados en SUMAFIJO se obtuvieron una serie de observaciones genéricas:

- El usar en reconocimiento una longitud asignada a la última neurona de 0 candidatos produce en todos los casos un mal comportamiento. En el resto de casos analizados (longitud asignada de 2500, 5000, 7500 o 10000 candidatos), dicho valor supone un factor que *suaviza* el resultado final en el sentido de que introduce un incremento adicional en la longitud de lista que permite que el sistema obtenga mejores tasas sin penalizar excesivamente el esfuerzo medio obtenido.

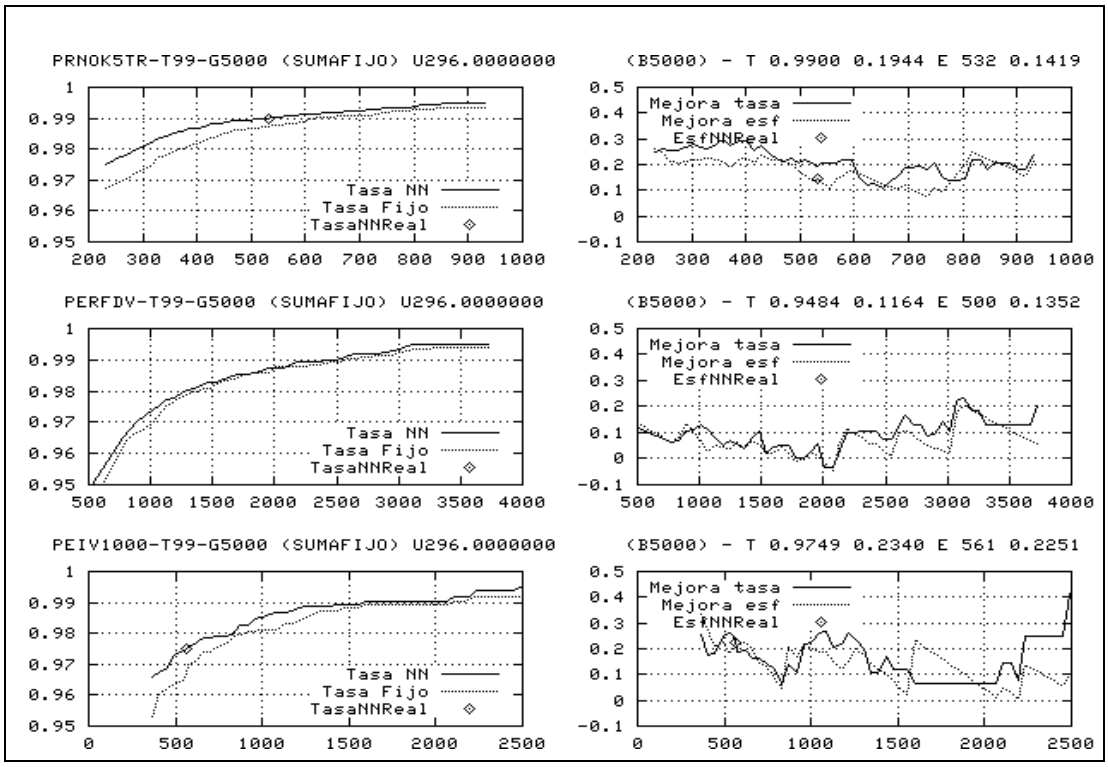

Figura 4-20: Resultados para SUMAFIJO ilustrando la dependencia con la longitud asignada al último segmento en reconocimiento (con T99)

- En **todos** los experimentos de interés (los que obtienen tasas de inclusión por encima del 98%), las disminuciones relativas en tasa de error para PERFDV oscilaban alrededor del 7-8%, mientras que para PEIV1000, alrededor del 20-30%, lo que demuestra la bondad de los resultados obtenidos.
- En **más del 90%** de los experimentos realizados, el método de redes neuronales consiguió mejoras tanto en tasa como en esfuerzo. Las excepciones se produjeron al usar tasas objetivo iguales a 98% o 98'5%, lo cual es lógico si pensamos que la búsqueda de tasas tan ajustadas en estimación (sobre el entrenamiento) al objetivo buscado en evaluación, combinadas con longitudes tan grandes llevaban a un umbral estimado mínimo, insuficiente para producir la generalización buscada.
- En **todos** los experimentos de interés, se obtienen mejoras relativas en tasa y esfuerzo en todo el rango de variación estudiado, lo que muestra la estabilidad de los resultados.
- El método basado en redes ha demostrado que es capaz de conseguir llegar a cumplir el requisito de tasa perseguido, 98%, con las siguientes matizaciones:
  - Para la lista PEIV1000, la mejora es más que notable, incluso manteniendo el esfuerzo medio por debajo del 10% del tamaño del diccionario, consiguiendo mejoras muy importantes en ciertos casos (de hasta el 40'74% en tasa para, por ejemplo, el experimento mostrado en la Figura 4-21, que también obtiene la mejor mejora relativa en disminución de error para la lista PERFDV)
  - Para la lista PERFDV, también se ha llegado al 98%, pero a costa de incrementar el esfuerzo medio por encima de ese 10% del que hemos hablado. El motivo es explicable si atendemos a la mayor dificultad de esta lista, como se verifica analizando las curvas de tasa de inclusión obtenidas en los métodos basados en listas fijas. El hecho de llegar a superar ese 98% en PERFDV incrementando el esfuerzo medio, implica el uso de unos parámetros que llevan a que, para las mismas

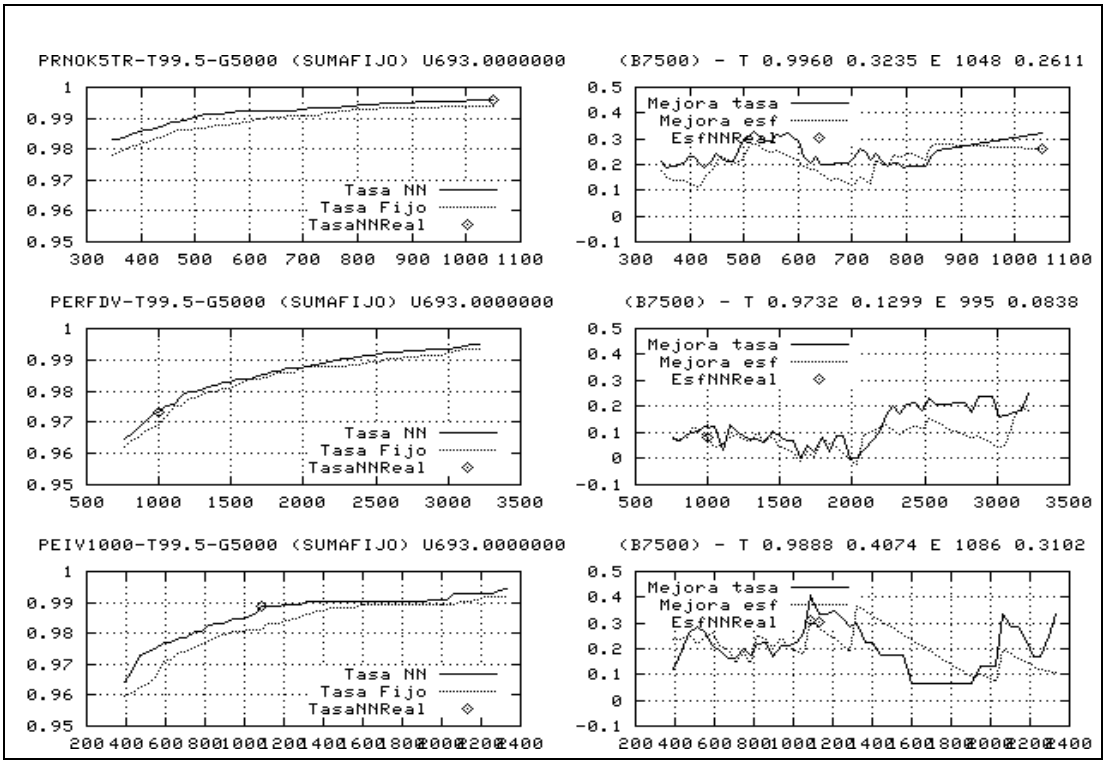

Figura 4-21: Resultados para SUMAFIJO con tasa objetivo del 99'5%, G=5000 y B=7500. Mejor resultado en disminución relativa de error para PERFDV y PEIV1000

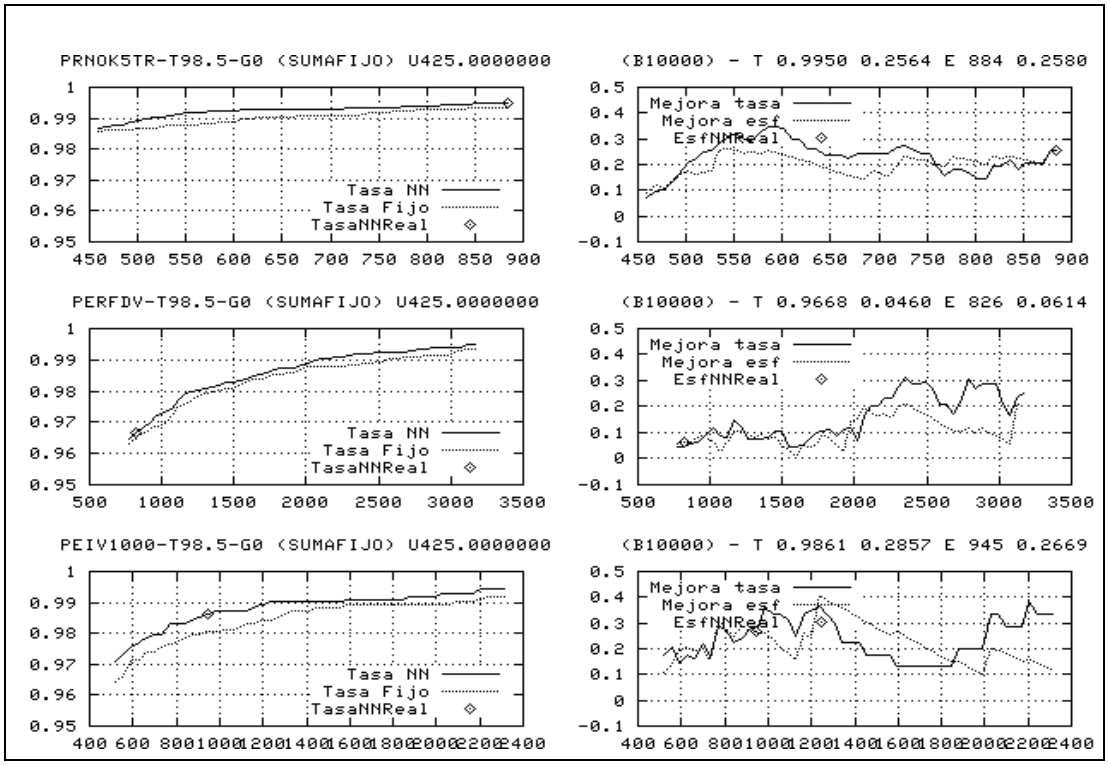

Figura 4-22: Resultados para SUMAFIJO con tasa objetivo del 98'5%, G=0 y B=10000

condiciones, se obtengan tasas mucho más elevadas en PEIV1000, como por ejemplo se muestra en la Figura 4-25, donde llegamos a un 99'09% en PEIV1000 y a un 98'36

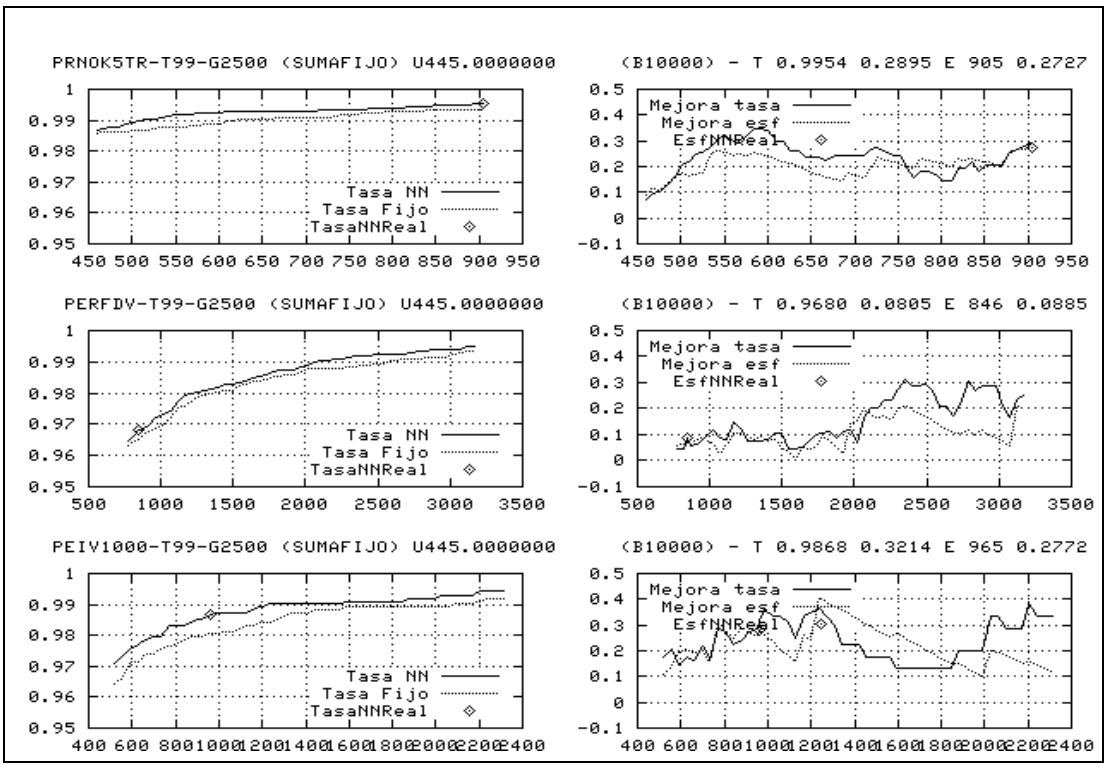

Figura 4-23: Resultados para SUMAFIJO con tasa objetivo del 99%, G=2500 y B=10000

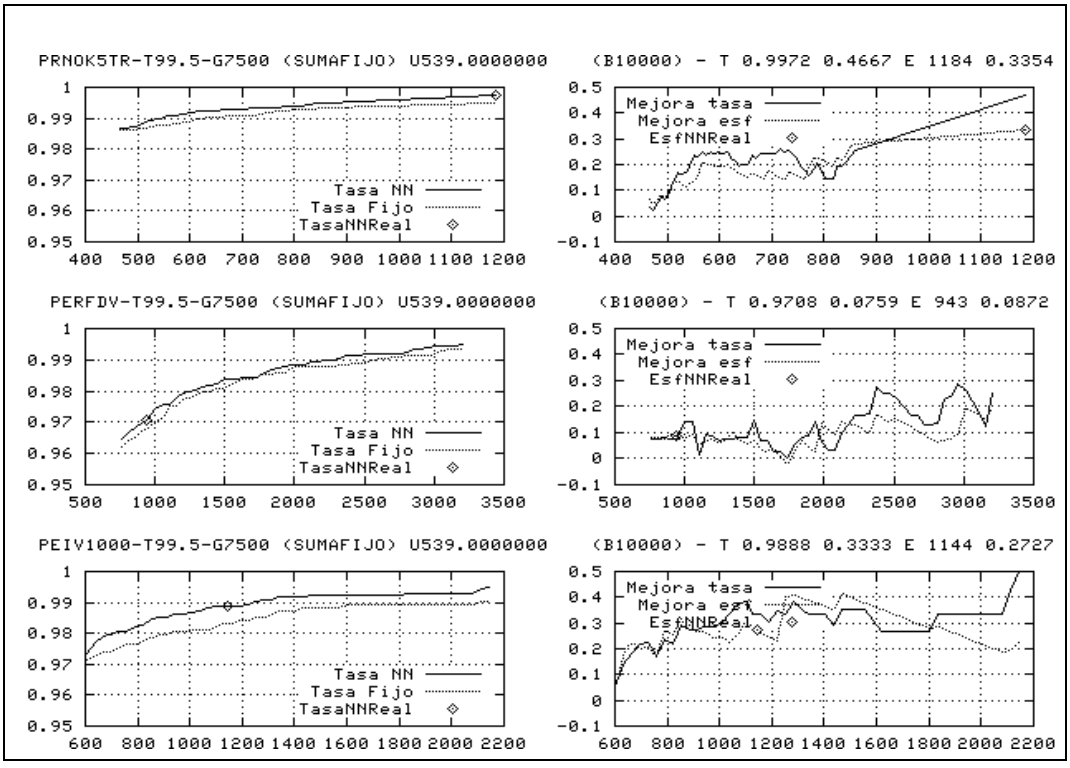

Figura 4-24: Resultados para SUMAFIJO con tasa objetivo del 99'5%, G=7500 y B=10000

- Si intentamos buscar criterios consistentes para seleccionar cuál de las alternativas usar (en cuanto a valores de G, T, B), puede decirse que:
  - El uso de un valor de B=10000 para la longitud asignada a la última neurona ofrece en todos los casos los mejores resultados.

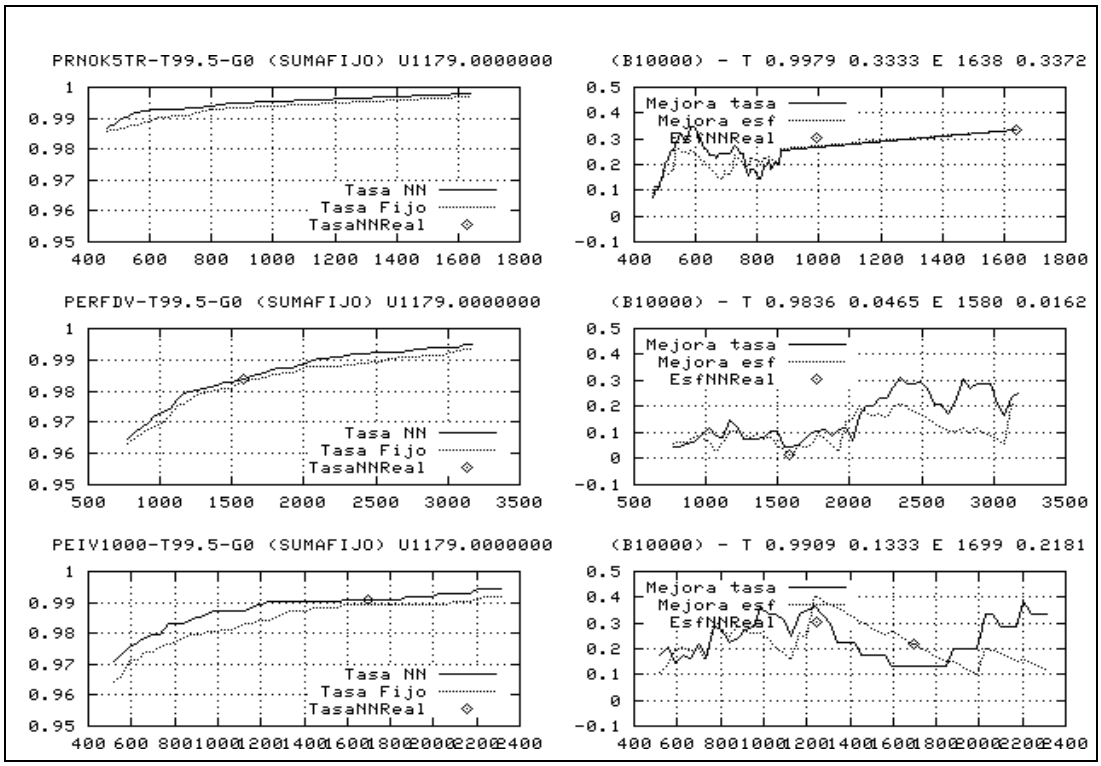

Figura 4-25: Resultados para SUMAFIJO con tasa objetivo del 99'5%, G=0 y B=10000. Mayor tasa obtenida para PEIV1000

- A medida que incrementemos el valor de G, es necesario incrementar el valor de la tasa objetivo a usar en entrenamiento, para tender a un umbral estimado que garantice un buen funcionamiento.
- Con estas pautas, podemos mostrar como ejemplo los resultados que aparecen en las figuras 4-22, 4-23, 4-24.

- La existencia de estas pautas consistentes y extraíbles a partir de los resultados, validan definitivamente la aproximación utilizada.

Tras los resultados vistos hasta ahora, las conclusiones fundamentales son:

- El método de estimación de longitudes de listas de preselección basado en el uso de redes neuronales ha demostrado ser una excelente alternativa al uso de listas fijas
- De las alternativas analizadas, la que utiliza post-proceso con el método SUMAFIJO es el que ha mostrado mejores resultados para todas las listas analizadas, siendo consistentes los resultados para un amplio rango de variación de los parámetros usados
- Es de destacar la notable mejoría relativa tanto en reducción de error como de esfuerzo que se ha obtenido en todos los experimentos realizados en los que se conseguía una tasa superior al 98% en PEIV1000, llegando a valores de entre un 20% y 30% para PEIV1000 y alrededor del 10% para PERFDV

El último aspecto que cabe considerar en este punto es hacer un estudio de bandas de fiabilidad para razonar sobre la validez estadística de los resultados. En las figuras 4-26, 4-27, y 4-28 se muestran dichos estudios para las tres listas consideradas y un subconjunto de los experimentos más relevantes[1]. Para cada par de barras, la de la izquierda se refiere al sistema basado en listas fijas y la de la derecha al basado en redes neuronales. La observación es clara: no nos es posible asegurar la

1. No es necesario visualizar todos ellos porque las observaciones son aplicables incluso en los casos no vistos en estas figuras. Los experimentos se identifican con un número del 1 al 10, y se incluyen los mismos experimentos para todas las bases de datos.

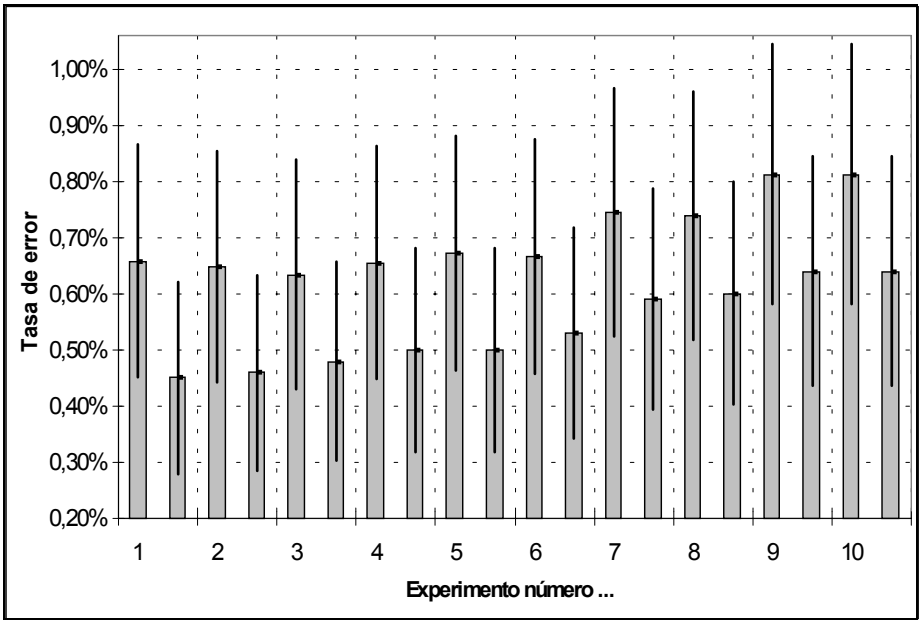

Figura 4-26: Bandas de fiabilidad para los experimentos más relevantes sobre la lista PRNOK5TR

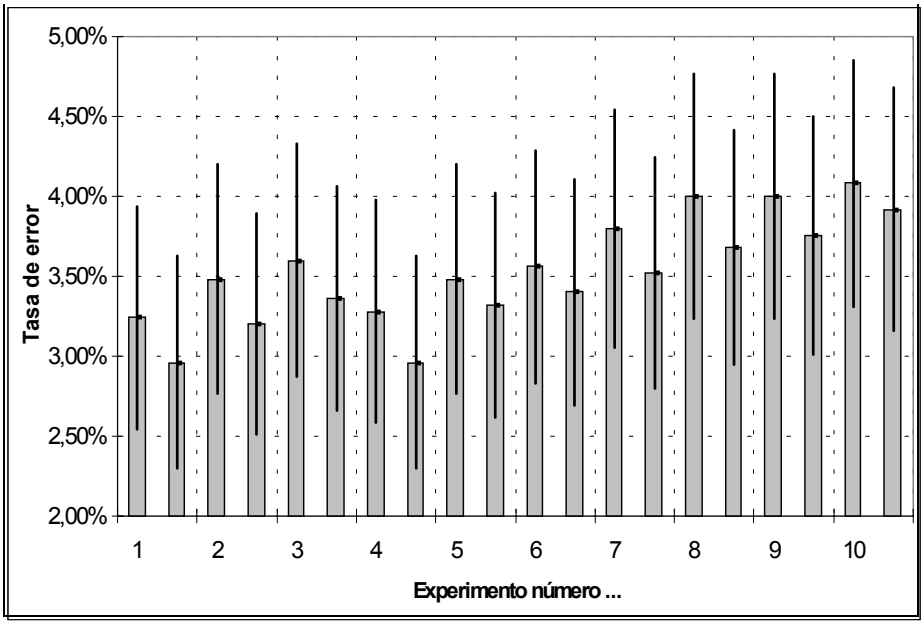

Figura 4-27: Bandas de fiabilidad para los experimentos más relevantes sobre la lista PERFDV

diferencia estadísticamente significativa de los resultados (con nuestra exigencia del 95% para dicha fiabilidad) dadas las limitadas bases de datos de que disponemos. Sin embargo, nuestra intuición y experiencia previa nos permiten confiar con razonable seguridad que de contar con bases de datos mayores, la diferencia observada será estadísticamente significativa. Baste como muestra considerar el hecho de la consistencia ya discutida de los resultados.

Por último queremos adelantar que los estudios de estimación de longitudes de listas con redes neuronales no concluyen con lo visto en este apartado, sino que continúan con el estudio del Apartado 4.5.3 "Uso directo de la activación de salida como estimador de longitud de lista". El motivo de retrasar su aparición hasta ese punto es la relación directa que tienen con los estudios de estimación de fiabilidad de hipótesis en cuanto a que fueron estos los que inspiraron aquellos.

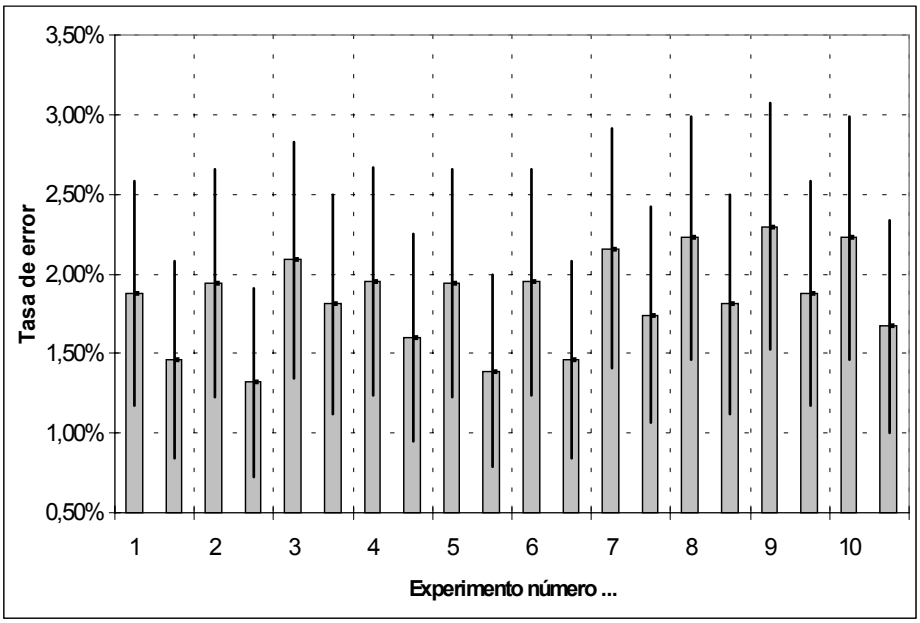

Figura 4-28: Bandas de fiabilidad para los experimentos más relevantes sobre la lista PEIV1000

4.5 Estimación de fiabilidad

Una de las mayores dificultades con las que se enfrentan los sistemas de reconocimiento de gran vocabulario es la estimación de hasta qué punto el resultado del proceso es o no fiable.

Dentro de los objetivos de la presente tesis se encontraba la búsqueda de una ligazón entre la estimación de longitudes de listas de preselección y la fiabilidad de las hipótesis presentadas.

Evidentemente ambos conceptos están relacionados: si conseguimos una estimación fiable de longitudes de listas de preselección y la longitud de lista estimada es pequeña, podremos asumir que la hipótesis del sistema es razonablemente fiable. En general, a mayor la longitud de la lista estimada, es previsible que sea más pobre la fiabilidad de la hipótesis del sistema.

Términos como longitud pequeña, razonablemente fiable, etc, son, como poco, ambiguos, de modo que nuestro planteamiento será intentar cuantificarlos y ofrecer soluciones prácticas que funcionen en sistemas reales.

Nuestro trabajo se centrará en utilizar algunos de los métodos planteados anteriormente para obtener una medida de la fiabilidad del reconocimiento. El objetivo final será usar dicha medida de fiabilidad como estimación directa de la longitud de lista a reconocer. Ese es el motivo que justifica la aplicación de estos métodos de estimación de fiabilidad al módulo de hipótesis con el que hemos venido trabajando hasta ahora, si bien no pensaremos en él como módulo de hipótesis, sino como módulo de reconocimiento en sí (verificación), distinguiendo el caso de reconocer cada palabra en la primera posición o en el resto.

4.5.1 Experimentos de discriminación

En los experimentos de discriminación descritos en el Apartado 4.4.9.7 a partir de la página 107, se tomaba la decisión de clasificación dependiendo de la activación de la neurona de salida. En el proceso tuvimos que decidir dónde poníamos el *umbral de decisión* que marcaba la frontera entre las activaciones *altas* o *bajas* de salida, optando por la posición media del rango de salida, es decir, en 0’5. Sin embargo, dicha decisión no tiene por qué ser la óptima, dado que no hemos tenido en consideración ningún criterio objetivo para fijar dicho valor.

En este apartado describiremos las experiencias desarrolladas usando una red neuronal como mecanismo de estimación de fiabilidad de hipótesis, con la estrategia planteada en los experimentos de discriminación descritos más arriba.

4.5.1.1 Análisis estadístico previo

En nuestro caso usaremos la red neuronal descrita en los apartados de estudios previos de potencia discriminativa que, recordamos, era entrenada para generar un valor de activación alto si la palabra había sido reconocida en posiciones distintas de la primera y viceversa. Para simplificar la descripción que se va a plantear, asumiremos que la red nos da un estimador de la probabilidad a posteriori de que una palabra dada haya sido reconocida en posiciones distintas de la primera. Así, entenderemos por aceptación (palabra acertada) el hecho de que la palabra haya sido reconocida en primera posición (valor bajo de activación) y rechazo (palabra fallada) el que haya sido reconocida en posiciones superiores a la primera (valor alto de activación).

El primer estudio a hacer atiende a las distribuciones de activación de los conjuntos de datos que pertenecen a cada categoría: palabras reconocidas en primera posición y palabras reconocidas en posiciones superiores a esa. En las figuras 4-29, 4-30 y 4-31 se muestran las distribuciones calculadas[1] de los valores de activación para cada conjunto, para las listas PRNOK5TR, PERFDV y PEIV1000, respectivamente (usando un diccionario de 10000 palabras). La observación más evidente es que es factible hacer una separación de cada conjunto basándonos únicamente en el valor estimado, es decir, la red tiene un buen comportamiento como estimador de fiabilidad[2] (a pesar incluso de que la superficie de error es grande), sobre todo si lo comparamos con estudios de la literatura que utilizan estimadores basados en probabilidades (o verosimilitudes) acústicas, para los que las distribuciones de cada conjunto de datos presentan un solape más acusado. Es también importante señalar que el punto de *equal error rate* (*EER*) está muy próximo al valor medio del umbral, 0'5

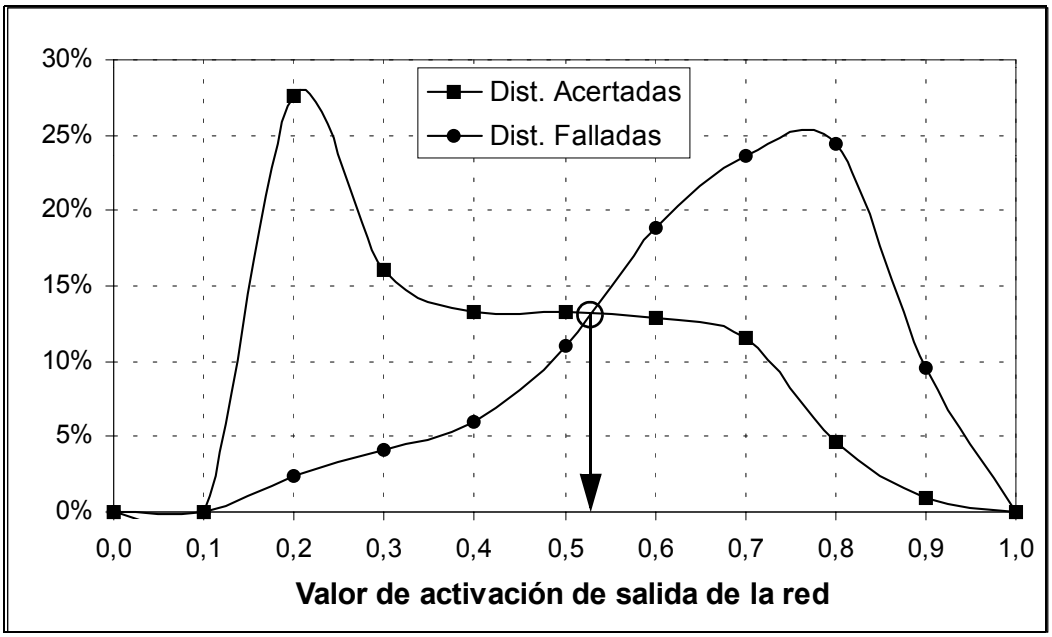

Figura 4-29: Histograma de activaciones para palabras acertadas y falladas para la lista PRNOK5TR

1. La estimación de las distribuciones se ha hecho de forma directa, es decir, por conteo de casos. Este método tiene el inconveniente de los errores introducidos por el muestreo. Métodos más precisos como el de la ventana de Parzen no son necesarios para nuestros objetivos.
2. De hecho, los estudios de correlación lineal entre la activación obtenida y la salida ideal muestran un valor alto del coeficiente de correlación

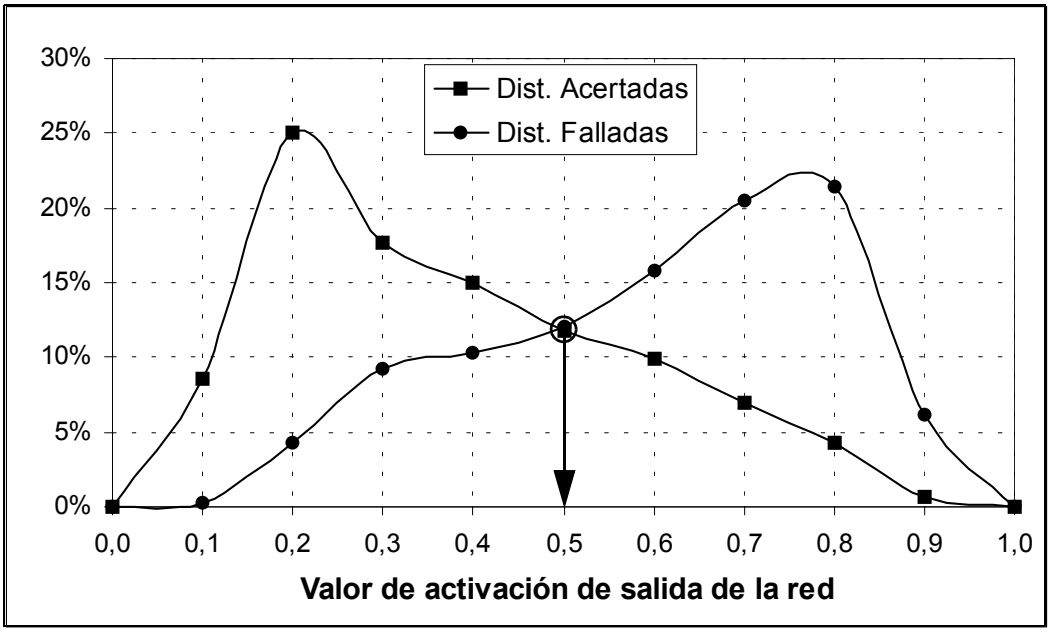

Figura 4-30: Histograma de activaciones para palabras acertadas y falladas para la lista PERFDV

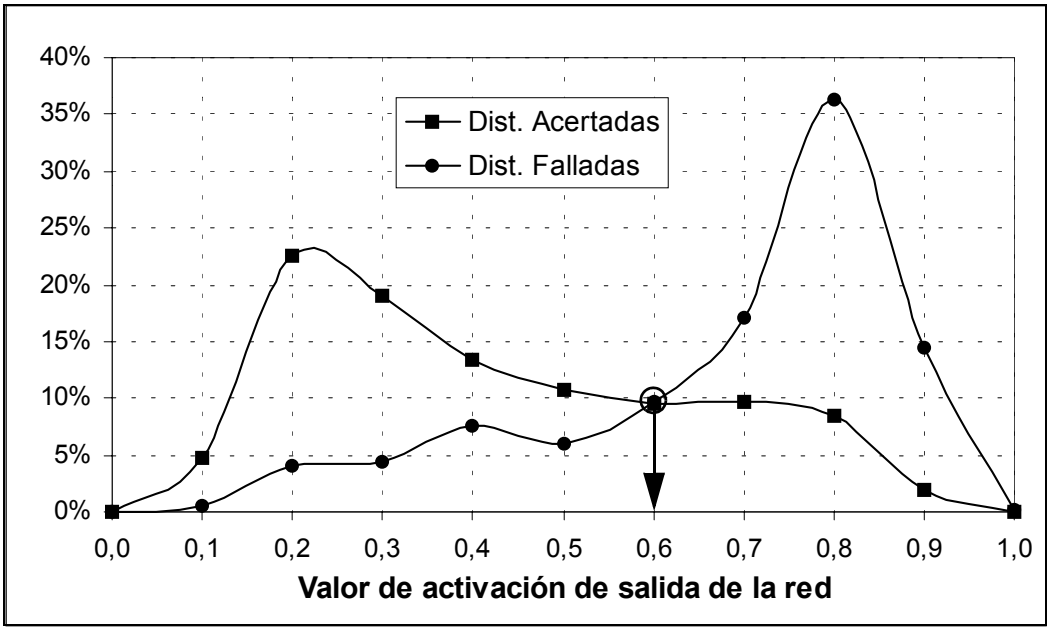

Figura 4-31: Histograma de activaciones para palabras acertadas y falladas para la lista PEIV1000

En el caso de redes neuronales, el método tradicional para diseñar una que decida entre dos posibles categorías[1] es conseguir un conjunto de ejemplos de cada tipo y entrenarla de modo que se obtenga un valor bajo de activación para una hipótesis y uno alto para la otra.

A la hora de tomar una decisión, ésta se basa en la comparación de la activación obtenida por la red y un umbral predefinido que, como decíamos antes, intuitivamente se puede colocar a la mitad del rango de activaciones fijadas para cada hipótesis, o más precisamente, donde se cortan ambas distribuciones (punto de *EER*), muy próximo a ese valor medio.

1. En nuestro caso hablamos de palabra correctamente reconocida (en posición 1 de la curva de tasa de inclusión) o no (en posiciones por encima de la primera).

4.5.1.2 Estimación de fiabilidad de hipótesis y errores en función del umbral utilizado

El estudio inmediato a realizar parte del cálculo de los errores posibles: La tasa de falsos rechazos (palabras correctamente reconocidas en primera posición pero detectadas como reconocidas en posiciones superiores a ésta) y la tasa de falsas aceptaciones (palabras reconocidas en posiciones superiores a la primera pero detectadas como reconocidas en primera posición), modificando el valor del umbral usado en el proceso de discriminación.

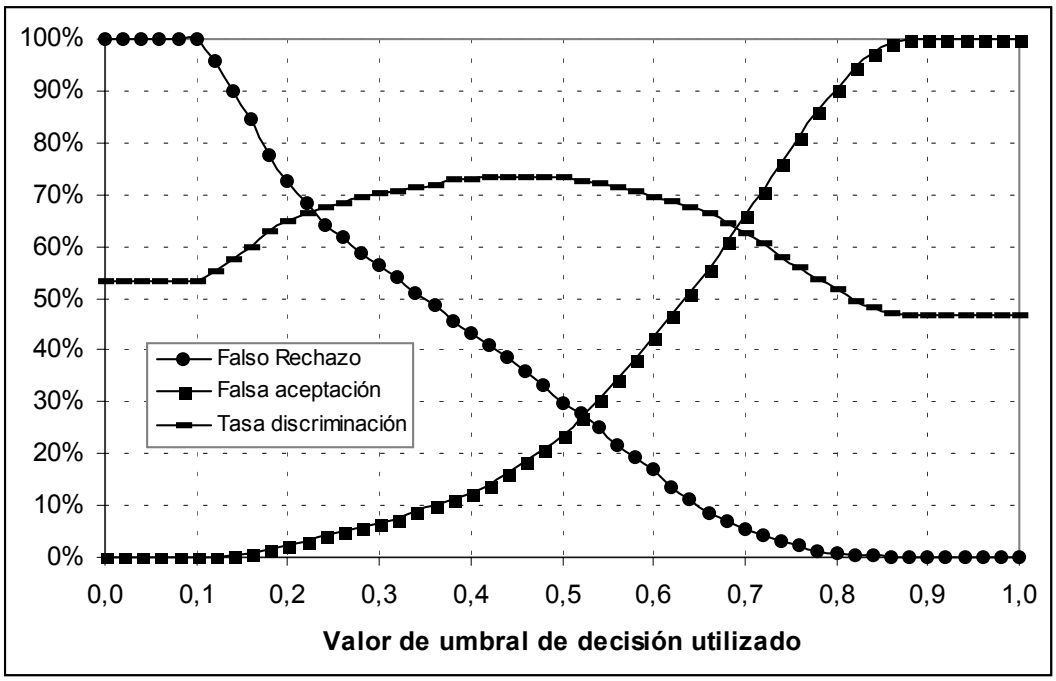

Figura 4-32: Valores de falsas aceptaciones, falsos rechazos y tasa conjunta de discriminación en función del umbral utilizado para PRNOK5TR

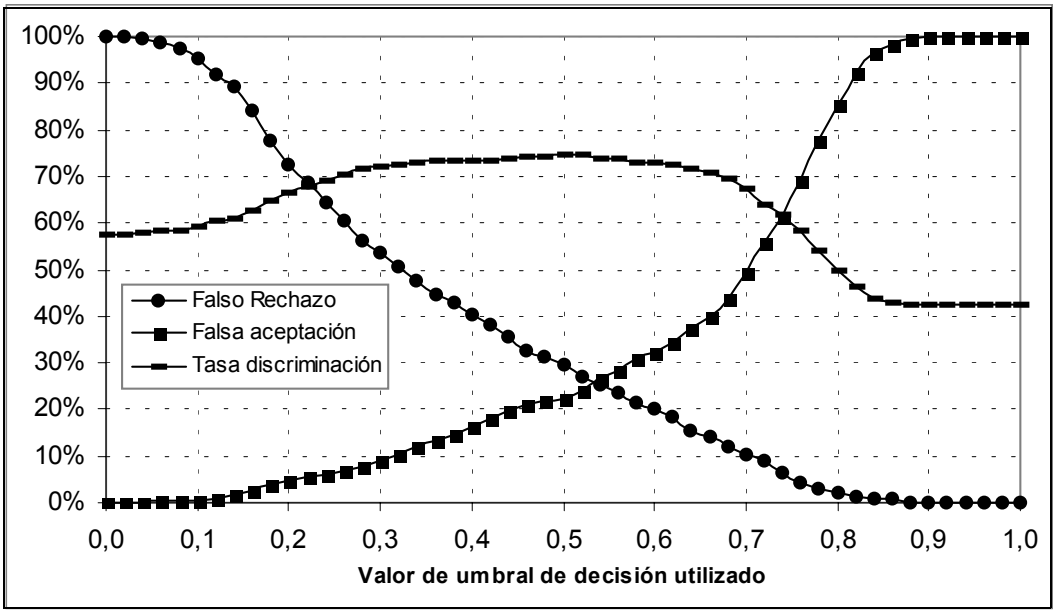

Figura 4-34: Valores de falsas aceptaciones y falsos rechazos en función del umbral utilizado para PEIV1000

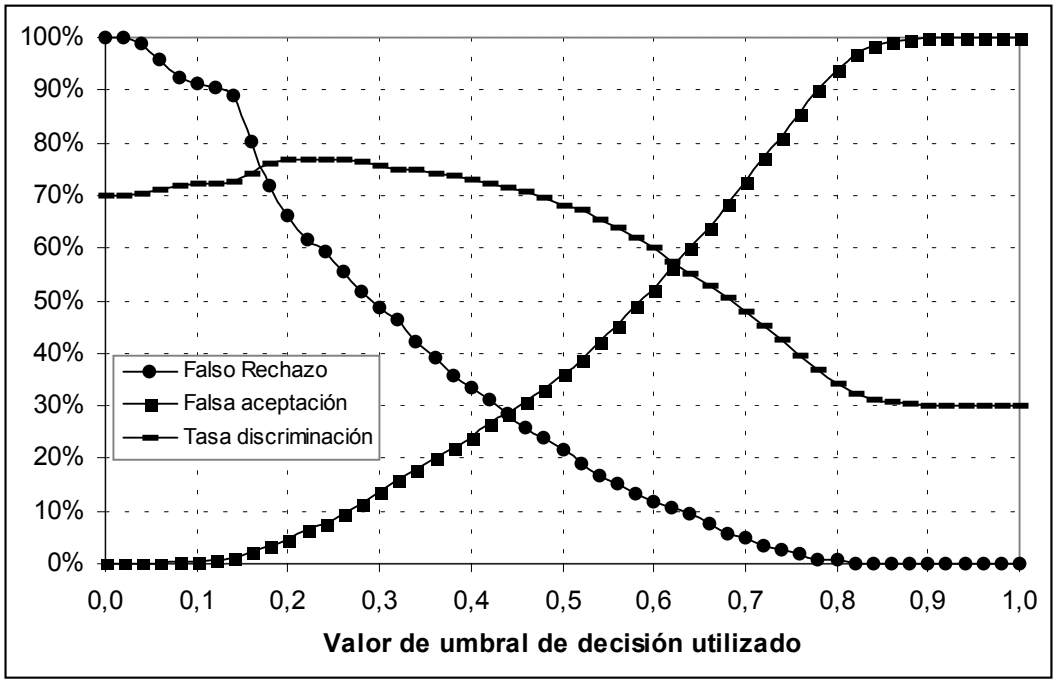

Figura 4-33: Valores de falsas aceptaciones, falsos rechazos y tasa conjunta de discriminación en función del umbral utilizado para PERFDV

En las figuras 4-32, 4-33 y 4-34 se muestran estos dos valores en función del umbral utilizado en la discriminación, para las listas PRNOK5TR, PERFDV y PEIV1000, respectivamente. El análisis de las mismas muestra las siguientes observaciones importantes:

- La decisión acerca del uso de un umbral en la mitad del rango disponible (es decir, de 0'5) es adecuada si buscamos una situación en la que penalizamos de la misma manera las falsas aceptaciones que los falsos rechazos. Las tasas de acierto obtenidas rondan el 70-75%, y los errores de ambos tipos el 25-30%.
- El comportamiento de la red para las tres listas es similar, en cuanto a tasas y a puntos de cruce de las curvas de error, lo que confirma la validez de la red como estimador sobre listas no conocidas en entrenamiento.
- Las tasas de discriminación alcanzables superan las tasas de reconocimiento para el primer candidato de los sistemas de partida (que constituyen las probabilidades "a priori" de la clase de palabras acertadas), lo que valida la utilidad del estimador de fiabilidad usado (por ejemplo, la tasa de acierto para el primer candidato en PEIV1000 es de un 42% y la tasa de discriminación en el punto de EER es de casi un 75%)
- Si analizamos la aplicabilidad práctica de un sistema de este tipo en una tarea de verificación, no de hipótesis, estudiando el efecto de admitir una cierta tasa de falsos rechazos, veremos que podemos llegar a incrementar notablemente la tasa de rechazos correctos. En la Tabla 4-15 se muestra la tasa de rechazos correctas para dos valores concretos de tasa de falso rechazo[1]. En ella puede verse como rechazando únicamente un 5% de las palabras correctas (es decir etiquetándolas erróneamente como falladas), podemos rechazar correctamente hasta prácticamente un 30% de las palabras falladas en PERFDV y hasta casi un 35% para PEIV1000. Si pensáramos en términos de tasa, eso querría decir que, caso de poder recuperar todas las palabras erróneas en una segunda pronunciación, por ejemplo, podríamos incrementar la tasa de reconocimiento para el

1. Nos interesa limitar el valor de falsas aceptaciones a tasas bajas porque es el error más perjudicial en nuestro sistema, como se discutió anteriormente.

primer candidato justamente en ese valor de rechazos correctos conseguidos (insistimos en que estamos pensando ya en un sistema de verificación en sí, no como módulo de preselección).

Tabla 4-15: Valores de Rechazo correcto para valores de falso rechazo (FR) dados

LISTA	*PARA FR=5%*	*PARA FR=2'5%*
PRNOK	31'86%	21'00%
PERFDV	29'79%	17'04%
PEIV1000	34'70%	22'25%

Tras esta discusión, es planteable retomar el tema de la estimación de longitudes de listas de preselección, es decir: ¿es factible usar los resultados del estimador de fiabilidad usado (del discriminador basado en redes neuronales) para extraer directamente una longitud de lista de preselección a usar? La diferencia con lo visto en el apartado Apartado 4.4.9.8, es que la red es ahora notablemente más simple y de probada capacidad discriminadora. Este enfoque es objeto de estudio del Apartado 4.5.3, pero antes describiremos los experimentos sobre fiabilidad para la tarea POLYGLOT.

4.5.2 Experimentos de discriminación para la tarea POLYGLOT con el sistema no integrado

Los buenos resultados obtenidos con las redes neuronales y los estimadores seleccionados como mejores en la tarea de cálculo de longitudes variables de listas de preselección nos llevaron a evaluar su comportamiento en la otra tarea planteada en esta tesis.

La tarea sobre POLYGLOT con el diccionario de 2000 palabras presenta mejores tasas de reconocimiento (valores próximos al 85%) que la vista hasta ahora sobre VESTEL (valores próximos al 30% para PERFDV y al 42% para PEIV1000), todo ello para las listas de evaluación. El caso para las de entrenamiento es aún más dispar: casi un 95% de acierto en POLYGLOT para el primer candidato y un 56% para PRNOK5TR.

Los resultados de discriminación para la base de datos de evaluación en POLYGLOT se muestran en la Figura 4-35, donde puede verse cómo la mayor descompensación de la base de datos de entrenamiento produce un desplazamiento del punto de EER, aunque mantenemos un valor del mismo del orden del 18%.

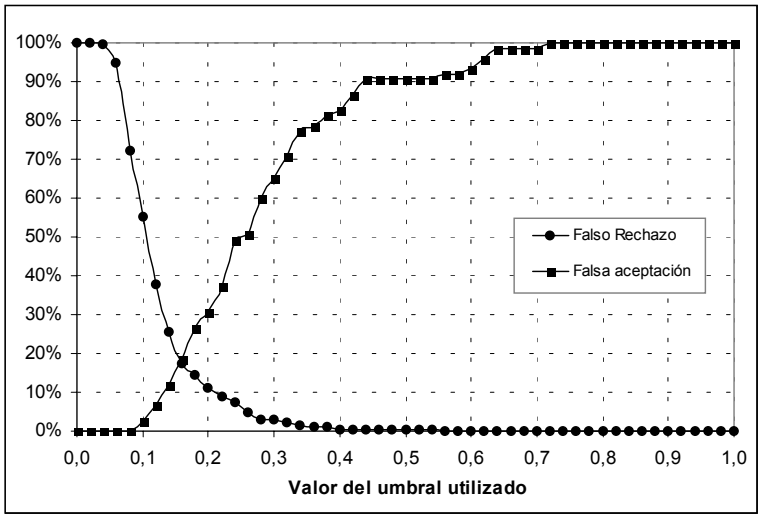

Figura 4-35: Tasas de falta aceptación y falso rechazo para la tarea POLYGLOT (base de datos de evaluación), usando el discriminador basado en redes neuronales con los 8 parámetros de entrada seleccionados.

En la Tabla 4-16 se muestran los valores de rechazo correcto para valores de falso rechazo admitidos de un 2'5% y 5%, y puede verse cómo los resultados son mejores incluso que los obtenidos para la tarea sobre VESTEL.

Tabla 4-16: Valores de Rechazo correcto para valores de falso rechazo (FR) dados sobre la tarea POLYGLOT

LISTA	*PARA FR=5%*	*PARA FR=2'5%*
set-c	49'7%	30'67%

4.5.3 Uso directo de la activación de salida como estimador de longitud de lista

El último estudio en este apartado se centró en la viabilidad del uso de la activación de salida de la red como un estimador directo de la longitud de la lista de preselección a usar.

Es discutible la inclusión de un estudio como este en un apartado titulado Apartado 4.5 "Estimación de fiabilidad", pero lo hemos dejado así al entender que usa un estimador de fiabilidad para lograr sus resultados.

Tal y como está concebido el uso de la red como discriminador, un valor alto de activación implicaría un valor alto de longitud de lista y viceversa.

Así, desarrollamos una serie de experimentos en los que se usaba un cálculo de longitud de lista proporcional a la activación de la red en cada palabra:

$$longLista(i) = actNN \cdot K$$

donde K es el factor de proporcionalidad a aplicar y sobre el que, de nuevo, hay que tomar una decisión.

La opción más evidente sería usar un valor de K proporcional a su vez al tamaño del diccionario. Por ejemplo, para un diccionario de 10000 palabras, un valor $K=10000$, con lo que una activación de 0'9 implicaría una longitud de lista de 9000 palabras, y un valor de 0'1 un valor de 1000[1]. Evidentemente esto podría dar lugar a una sobreestimación excesiva de longitudes, lo que produciría un esfuerzo medio elevado. Como nuestro objetivo secundario es bajar también de la cifra del 10% del tamaño del diccionario, hemos modificado el requisito para K, de modo que sea:

$$K = \frac{\text{TAMAÑO\_DICCIONARIO}}{\phi}$$

donde ϕ tiene un valor entre 1 y 9.

En estas condiciones, se midió la tasa de inclusión obtenida y el esfuerzo medio y se hizo una comparación con el caso de usar listas de longitud fija. En la Figura 4-36 se muestran las tasas de inclusión obtenidas para las tres listas procesadas, en la Figura 4-37 la reducción relativa de error (comparando con el sistema de listas fijas para un esfuerzo medio igual al fijo que consigue la misma tasa) y en la Figura 4-38 la reducción relativa en esfuerzo (para una tasa igual a la conseguida en el sistema de listas fijas con el mismo esfuerzo), todo ello en función de ϕ (eje de abscisas).

Lo más destacable es la consistencia de los resultados obtenidos para las tres listas procesadas. En todas ellas y para todos los valores de ϕ usados, se consiguen mejoras tanto en tasa como en esfuerzo medio. A la vista de la lista de entrenamiento, sería razonable usar un valor de ϕ=5, que produce un máximo local en reducción de error (Figura 4-37) alejado del extremo poco razonable

1. Es conveniente recordar aquí que la red se entrena usando valores de 0'1 y 0'9 para identificar cada caso (palabra reconocida en primera posición y resto, respectivamente).

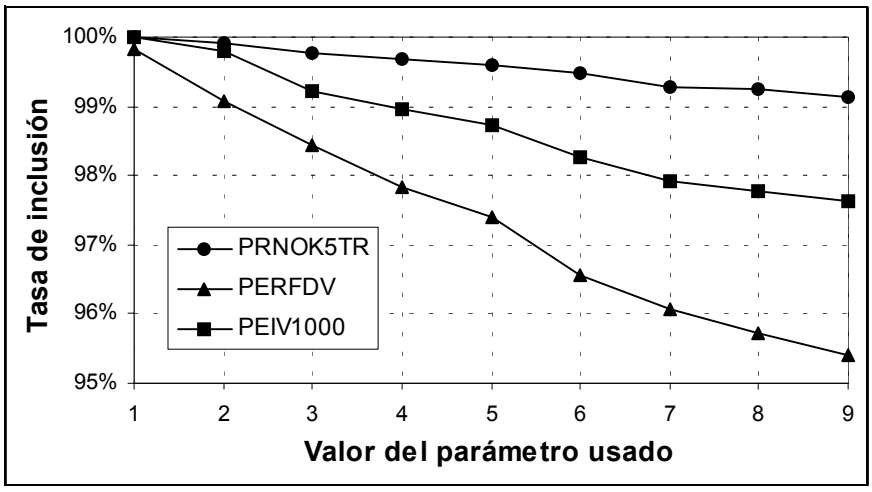

Figura 4-36: Tasas de inclusión para las tres listas en los experimentos de estimación de longitud de lista dependiente de la activación de la red (en función de ϕ)

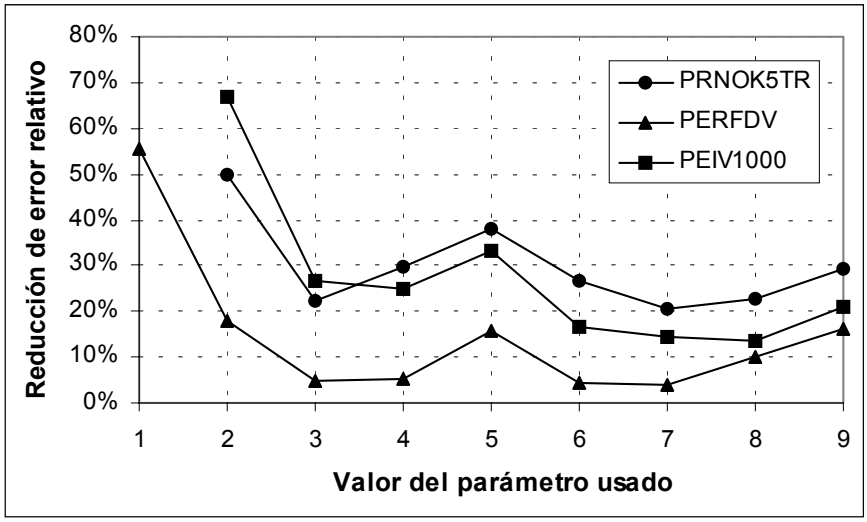

Figura 4-37: Reducción relativa de error de inclusión para las tres listas en los experimentos de estimación de longitud de lista dependiente de la activación de la red (en función de ϕ)

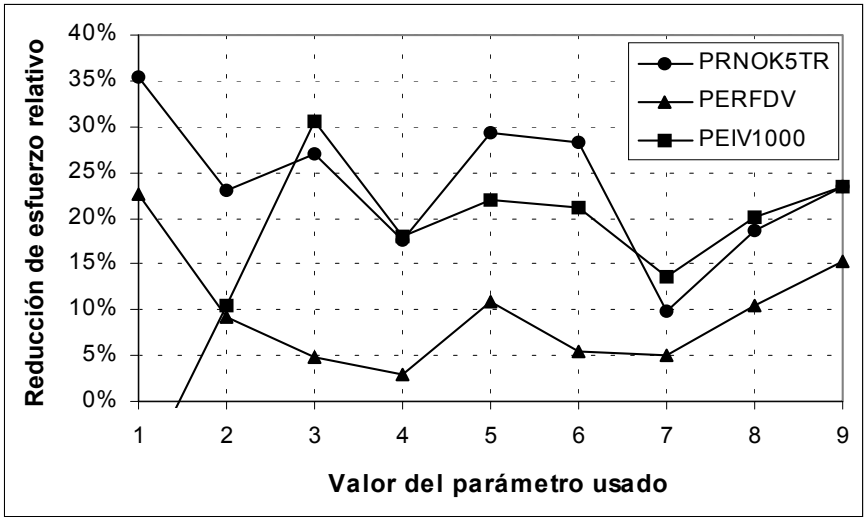

Figura 4-38: Reducción relativa de esfuerzo para las tres listas en los experimentos de estimación de longitud de lista dependiente de la activación de la red (en función de ϕ)

de $\phi=1$ (ya que sabemos positivamente que alcanzaremos la tasa del 100% mucho antes de 10000 candidatos). En este punto conseguimos tasas muy superiores al 98% para PEIV1000 (98'74% exactamente) y del 97'40% para PERFDV, manteniendo el esfuerzo muy ajustado al 10% del tamaño del diccionario que tenemos como objetivo (1043 candidatos para PEIV1000 y 972 para PERFDV).

Es relevante hacer notar cómo los resultados obtenidos son asimilables a los presentados en el Apartado 4.4.9.8, al hablar de la estimación de longitudes de listas usando una red neuronal de 10 salidas. Los comentarios que allí hacíamos respecto a la fiabilidad estadística de la diferencia entre los resultados son los mismos en estos experimentos, por lo que no volveremos a insistir en ellos.

La conclusión fundamental de este apartado vuelve a ser la consecución de mejores resultados que los experimentos con listas fijas al usar una red neuronal como estimador.

En el Apartado 4.4.9.8 discutíamos las conclusiones al usar un estimador completo con 10 salidas. En este apartado hemos diseñado y analizado el funcionamiento de una red mucho más cómoda de manejar y entrenar con prestaciones similares. Lo único que faltaría por abordar sería un estudio de sensibilidad tan completo como el visto allí, pero que entendemos no es necesario si pensamos en que las figuras mostradas responden precisamente a esta idea: hemos modificado el punto de trabajo afectando a un parámetro del sistema y las mejoras son constantes en todo el rango y consistentes entre todas las bases de datos analizadas.

Una ventaja adicional de este sistema es que tiene menos valores a estimar que la red más complicada, en cuanto a que no hay que calcular umbrales adicionales ni decidir acerca de la longitud a asignar a la última neurona de salida (tal y como se detallaba en el Apartado 4.4.9.8.3, al hablar de los parámetros de control de los experimentos).

4.5.4 Consideraciones sobre el uso o no de redes neuronales en estimación de confianza

En la literatura se pueden encontrar ejemplos del uso de redes neuronales para tareas de estimación de fiabilidad, además de otros tradicionales en los que se usa simplemente un parámetro (o combinación de ellos) para decidir.

En nuestro caso se hicieron experimentos utilizando redes neuronales o el valor directo del parámetro de entrada para discriminar. Los resultados obtenidos no son concluyentes en cuanto a que uno sea mejor que otro, porque las tasas obtenidas por ambas estrategias se solapan al aplicar el estudio de fiabilidad estadística, con lo que no está claro que la red neuronal sea mejor que el uso directo del parámetro.

Sin embargo, nuestra propuesta es sin duda el uso de la red neuronal, por su facilidad de cara a la integración de distintos parámetros y la facilidad de su aplicación a la tarea, sobre todo teniendo en cuenta que su funcionamiento será como mínimo igual al del enfoque tradicional.

4.5.5 Consideraciones sobre la evaluación y estimación del umbral de decisión

En la mayor parte de los estudios presentes en la literatura sobre estimación de fiabilidad de hipótesis (y en nuestra descripción hasta este punto) el interés se centra en el cálculo de potencia discriminativa, ofreciendo diversas medidas al respecto (EER, curvas ROC, curvas de falsas aceptaciones y falsos rechazos, etc.).

Sin embargo, muy pocas de ellas llegan a proponer una metodología concreta de estimación del umbral a usar pensando en la aplicación real del sistema.

Nuestra propuesta es utilizar como valores de estimación de umbral aquellos que podamos calcular a partir de los experimentos sobre la lista de entrenamiento, como pueden ser el punto en el que se obtiene el EER, o el mínimo error de clasificación, o el mínimo coste con la estrategia de selección que se haya decidido, o valores concretos de falso rechazo permitido FR (2'5% o 5%, por ejemplo).

De todos ellos, y para todos los experimentos realizados (para VESTEL y POLYGLOT), el criterio que mejor comportamiento obtiene (en el sentido de conseguir un mejor equilibrio entre falsos rechazos (preferiblemente bajos) y rechazos correctos (preferiblemente altos))[1] es el basado en aplicar el valor de umbral estimado para producir, sobre la lista de entrenamiento, un determinado falso rechazo. En la Tabla 4-17 se muestran los resultados sobre las bases de datos de evaluación para dos valores de umbral, obtenidos imponiendo un falso rechazo del 2'5% y del 5% en la lista de entrenamiento.

Tabla 4-17: Valores de Rechazo correcto (RC) para valores de falso rechazo (FR) dados sobre la tareas VESTEL y POLYGLOT usando el umbral estimado en entrenamiento para un valor de FR=2'5% y FR=5%

LISTA	*PARA FR=5% en entrenamiento*	*PARA FR=2'5% en entrenamiento*
set-c POLYGLOT	FR=7'33%, RC=50'67%	FR=2'36%, RC=29'33%
PERFDV	FR=4'25%, RC=25'16%	FR=2'45%, RC=17'04%
PEIV1000	FR=9'72%, RC=47'88%	FR=5'11%, RC=34'70

4.6 Conclusiones

En este capítulo se han presentados en primer lugar estudios sobre distintas estrategias de organización del espacio de búsqueda orientadas a exploración y búsqueda con algoritmos de programación dinámica: árboles y grafos, deterministas y no deterministas. Se han incluido cálculos reales de los ahorros producidos por cada uno de ellos, así como consideraciones sobre su utilidad en la búsqueda y los problemas asociados al uso de grafos (no optimalidad de la solución por la compresión excesiva del espacio de búsqueda y dificultad de aplicación en casos prácticos con buenos resultados). Se han analizado igualmente soluciones para incrementar la tasa de inclusión obtenible sobre estructuras de grafo, combinando de forma óptima la información disponible en el proceso, pero no se ha llegado a un rendimiento similar al obtenido con los árboles.

En lo que respecta a estrategias de reducción del esfuerzo computacional de sistemas basados en el paradigma hipótesis-verificación, se ha trabajado intensamente en la idea de estimación de listas variables de preselección, analizando de forma preliminar la aplicación de métodos paramétricos y no paramétricos (parte de esta investigación en profundidad queda planteada para trabajos futuros) y profundizando en el uso de redes neuronales como mecanismo estimador. Se ha presentado una propuesta ambiciosa de parámetros a considerar, de entre los disponibles en los procesos de preselección, proponiendo una metodología de selección de parámetros de entrada, topologías y métodos de codificación, en base a su potencia discriminativa, estimada también con redes neuronales, en una tarea simplificada. Se han abordado soluciones concretas al cálculo final de longitudes de lista a partir de la salida de la red y se ha hecho una amplia evaluación de los resultados de la estrategia propuesta, en la que se ha propuesto un mecanismo original de comparación con el enfoque tradicional de uso de listas de longitud de fija. Se ha mostrado la consistente mejora conseguible con el uso de redes neuronales, pero no se ha establecido de forma concluyente la fiabilidad estadística de las diferencias apreciadas, dadas las limitaciones en las bases de datos disponibles, lo que queda planteado para trabajos futuros.

El esfuerzo realizado en estimación de longitud de listas de preselección se ha extendido en su ámbito de aplicación, de forma natural, al problema de estimación de fiabilidad de reconocimiento,

1. Estamos pensando de nuevo en la aplicación como sistema de reconocimiento, no de preselección.

obteniendo buenos resultados tanto en VESTEL como en POLYGLOT. Se han mostrado igualmente las posibilidades de incremento de la tasa *subjetiva*[1] de reconocimiento del sistema admitiendo una cierta tasa de falsos rechazos, lo que abre interesantes posibilidades para esta aplicación y que deberán ser validadas con un estudio similar sobre sistemas con tasas más elevadas de reconocimiento en el primer candidato, obviamente.

En este campo se han hecho igualmente estudios acerca de la bondad del uso de redes neuronales frente al uso típico de estimadores directos, habiéndose demostrado que si bien se obtienen resultados similares en ambos casos, las redes ofrecen una ventaja competitiva importante por su facilidad de uso y su capacidad de integrar un número elevado de parámetros de entrada sin esfuerzo. Igualmente se ha discutido el problema de la evaluación y estimación del umbral de decisión de cara a su uso en sistemas reales, proponiendo criterios de selección de dicho umbral.

Finalmente, se ha vuelto a aplicar la idea de estimación de fiabilidad, de forma directa, a la estimación de longitudes de lista, obteniendo excelentes resultados, comparables a los de las estrategias más complejas planteadas en este capítulo.

1. Con *subjetiva* nos referimos a que no es factible mejorar la tasa real del sistema: si una palabra se falla, fallada está. Sin embargo, podríamos solicitar al usuario la repetición de la palabra y, eventualmente, reconocerla correctamente. La tasa no se ha mejorado, pero el sistema ofrece al usuario la visión de que sabe que no le ha entendido y solicita dicha repetición.

5 Selección de unidades y diccionarios

En este capítulo se estudiarán y desarrollarán soluciones a dos problemas fundamentales en la construcción de sistemas de reconocimiento de habla:

- La determinación del repertorio de unidades básicas de reconocimiento, a lo que se dedicará la primera parte, junto con la experimentación asociada
- La determinación de las entradas de los diccionarios a usar, con la introducción de múltiples alternativas de pronunciación, a lo que se dedicará la segunda parte

con el objetivo en ambos casos de introducir criterios objetivos de diseño y evaluación del impacto del mismo, todo ello a la luz de las discusiones arquitecturales del Capítulo 3.

Igualmente se evaluará el impacto en el rendimiento del sistema de la introducción de distintos tipos de modelado, alfabetos, etc., y se discutirán conceptos relacionados con la independencia del vocabulario y la evaluación de la complejidad de diccionarios.

5.1 Selección de unidades

5.1.1 Modelado

A lo largo de la tesis se han analizado distintas alternativas de modelado, todas ellas basadas en el uso de HMMs, bien discretos o semicontinuos.

- Modelos discretos de Markov (DDHMM, *Discrete Densities Hidden Markov Models*): Su ventaja fundamental es el reducido coste computacional que requieren, adoleciendo sin embargo de una falta de precisión en el modelado notable, especialmente en tareas difíciles (independencia del locutor y gran vocabulario). En la actualidad su uso se orienta a sistemas de pequeño vocabulario y dependencia del locutor (aquellos en los que la variabilidad acústica es lo suficientemente reducida como para garantizar un funcionamiento adecuado) o bien en aquellos módulos en los que se busca un coste computacional reducido y se dispone de módulos posteriores con mejor capacidad de modelado que permitan conseguir tasas de reconocimiento superiores.
- Modelos semicontinuos de Markov (SCDHMM, *SemiContinuous Densities Hidden Markov Models*): Más costosos que los anteriores, pero sin llegar a las demandas computacionales de los sistemas continuos, presentan un balance adecuado teniendo en cuenta su capacidad de modelado. Este modelado se puede ver de diferentes maneras: como un modelo discreto precedido de una cuantificación suave (*soft quantization*) o como un modelo continuo en que se comparten las gausianas de los diferentes modelos (*parameter tying*). De hecho, el modelado semicontinuo es un punto intermedio entre el discreto y el continuo, que pese a ser teóricamente menos potente que el continuo, con las limitaciones de material de entrenamiento típicas, se acerca bastante al rendimiento que se puede obtener en esas mismas condiciones de un modelado continuo. En nuestro caso, y a partir de las conclusiones obtenidas en estudios previos en nuestro Grupo [Córdoba95, Ferreiros96], sólo nos quedamos con los cuatro mejores valores de verosimilitud. Por otra parte, trabajamos en aritmética entera, usando log-verosimilitudes (*log-likelihoods*) y la técnica de suma de log-verosimilitudes que se describe en [Ferreiros96] .

Los modelos continuos de Markov (CDHMM, *Continuous Densities Hidden Markov Models*) no han sido objeto de estudio porque el material inicial de entrenamiento con el que se contaba no permitía, a priori, garantizar un adecuado entrenamiento de todos los parámetros necesarios para obtener mejoras significativas con relación a los semicontinuos, aunque pueden consultarse los trabajos previos al respecto sobre VESTEL y SPEECHDAT en [Gaviña00] y [Moreno01].

En todos los casos se ha usado la misma topología para modelar alófonos, la del modelo de Bakis tradicional [Bakis82]: modelos de tres estados, con transiciones simples y dobles de izquierda a derecha, como aparece en la Figura 5-1. A lo largo del desarrollo de esta tesis y dentro del Grupo de

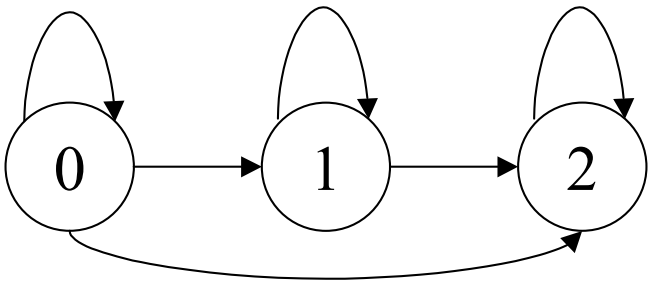

Figura 5-1: Topología de los HMM usados

Tecnología del Habla se plantearon topologías alternativas que produjeron mejores resultados en tareas concretas [Moreno01], pero se desestimó su uso por no ser objeto de nuestro estudio profundizar en ese aspecto.

Como se ha comentado, la selección de unidades a entrenar es uno de los problemas fundamentales a solventar en el diseño de sistemas de reconocimiento automático de habla. Las opciones a plantear buscan distintos grados de compromiso entre (fundamentalmente[1]):

- Consistencia: Las distintas repeticiones disponibles de cada unidad deben tener características similares
- Entrenabilidad: Debemos contar con un número suficiente de repeticiones de cada una como para asegurar un entrenamiento robusto y correcto de los modelos asociados

En el caso de los modelos alofónicos, es posible conseguir un número suficiente de repeticiones en casi todos los casos[2], pero no son del todo consistentes entre sí dados los fuertes efectos de coarticulación que se dan entre sonidos adyacentes.

Una solución típica al problema de la consistencia de los alófonos consiste en utilizar modelos superiores, siendo el uso de los tri-alófonos la estrategia más común.

5.1.2 Entrenamiento de modelos

El proceso que sigue el sistema de entrenamiento recibe a su entrada la secuencia de observaciones acústicas correspondientes a cada palabra y debe conocer, además, la secuencia de unidades que la componen (usando un sistema de conversión de grafema a fonema). Está basado en el tradicional *Segmental K-means* [Rabiner89a], y no lo describiremos en profundidad. Baste decir que se construye el modelo correspondiente a cada palabra concatenando los modelos individuales de las unidades que lo forman, como aparece en la Figura 5-2. En nuestro caso, el modelo *0* sería el de silencio inicial, y el modelo *n* el de silencio final. En la implementación se ha permitido igualmente el salto doble desde cualquier estado, lo que da mayor libertad en las fronteras entre unidades. Sobre este modelo se ejecuta el algoritmo de Viterbi, del que se obtiene una segmentación determinada y una verosimilitud correspondiente a esa segmentación.

A partir de la segmentación se reestiman las matrices *a* y *b* por el método tradicional de máxima verosimilitud, y todo el proceso se repite para todas las palabras y para todas las unidades mientras la verosimilitud de Viterbi converja adecuadamente, siguiendo una estrategia EM. Las

1. En el capítulo de encuadre científico-tecnológico se puede encontrar una descripción más amplia del problema y alternativas presentes en la literatura.
2. A este respecto puede consultarse el Anexo D donde se ofrecen datos cuantitativos sobre el número de repeticiones disponibles en las bases de datos en función del alfabeto utilizado.

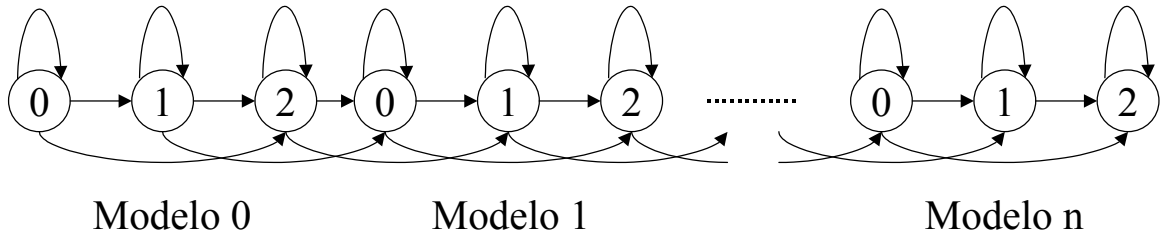

Figura 5-2: Topología de los HMM usados en entrenamiento como concatenación de unidades

semillas iniciales que se usan en la primera iteración para estimar los HMM discretos se obtienen por segmentación inicial equidistante, ya que se ha demostrado que el hecho de elegir una inicialización aleatoria u otra, o incluso una fija, no influye significativamente en el rendimiento del sistema (manteniéndose la variación del mismo por debajo del 1% [Rabiner83], lo que ha sido verificado en trabajos previos de nuestro Grupo).

El entrenamiento de modelos semicontinuos es una variación del caso de los discretos, en los que inicializamos el proceso con dichos modelos discretos, añadiendo las varianzas de los centroides correspondientes al cuantificador vectorial usado en el modelado discreto. El proceso está descrito en detalle en [Ferreiros96].

En lo que respecta al entrenamiento de los modelos de tri-alófonos, se introduce un proceso adicional de agrupamiento, dados los problemas de entrenamiento con los que nos encontramos para algunos de ellos, que describiremos en el Apartado 5.1.4).

Para suavizar los parámetros de los HMM estimados, la técnica de suavizado umbral ha sido la elegida en nuestro caso por su simplicidad, aunque dada su incapacidad de distinguir los símbolos de salida improbables de los imposibles, la aplicamos introduciendo una sencilla modificación que permite eliminar los ceros y los valores demasiado pequeños de las matrices, al tiempo que favorecemos por igual la probabilidad de ocurrencia de todas las observaciones (básicamente se trata de sumar el valor de probabilidad umbral a *todas* las probabilidades, en lugar de únicamente a las no entrenadas, y normalizar a continuación).

5.1.3 Selección manual de unidades independientes del contexto

Para abarcar un rango razonable en la selección de unidades a modelar y teniendo en cuenta la disparidad de los sistemas y tareas sobre los que los aplicaremos (integrados y no integrados, tareas de dependencia e independencia del locutor), se diseñaron cuatro alfabetos distintos, tal y como se describe en el Anexo D, con un número de unidades que varía entre las 51 del más completo (`alf51`) hasta las 23 del más sencillo (`alf23`), pasando por las 33 y 45 de dos alfabetos intermedios (`alf33` y `alf45`). En cada caso, los criterios de selección son fundamentalmente lingüísticos y los que atienden a la consecución de un buen compromiso entre entrenabilidad adecuada y resolución acústica, lo que se combina con consideraciones que atienden a las ocurrencias disponibles en la base de datos de entrenamiento.

La generación de diccionarios en los que se incluye la descomposición de cada entrada en unidades elementales se hace a partir del conversor grafema-fonema de nuestro Grupo, al que se le aplican filtros de transformación en los alfabetos más sencillos.

En trabajos como los de [Hassan90] y [Ferreiros96] se proponen modificaciones adicionales al alfabeto completo (`alf51`), orientadas a conseguir mayor resolución de modelado acústico en símbolos concretos (como por ejemplo las oclusivas). En este trabajo no se han usado por no considerarlo necesario dadas las arquitecturas de trabajo y debido a que las conclusiones de aquellos no son directamente extensibles a nuestro caso, dada la diferencia fundamental en filosofía de diseño, por lo que debería ser objeto de un estudio en profundidad.

En cualquier caso, el sistema es lo suficientemente flexible como para poder especificar alfabetos distintos con toda generalidad, de modo que pruebas con los mismos sean inmediatas.

5.1.4 Selección automática de unidades dependientes del contexto

De cara a incrementar la tasa de reconocimiento en las tareas más complicadas que se plantearon en esta tesis, se decidió abordar el uso de unidades dependientes del contexto, dado que había experiencia previa en el Grupo que aseguraba que era posible conseguir un adecuado entrenamiento con el material acústico disponible para este fin.

El esquema de generación es similar al descrito en [Córdoba95] y [Ferreiros96]. Básicamente se parte del conjunto completo de tri-alófonos presentes en la base de datos de entrenamiento, inicializándolos a partir de los modelos independientes del contexto correspondientes al alófono central. A partir de ahí se aborda un proceso de agrupación de las distribuciones entrenables. La aproximación más sencilla parte de realizar el agrupamiento considerando únicamente la cantidad de datos de entrenamiento disponibles para cada unidad y un conjunto de umbrales [Lee89][Lee90] pero en nuestro caso nos apoyamos en trabajos previos del Grupo en los que se analizaron distintas estrategias de agrupamiento [Córdoba95], optando finalmente por agrupamiento de distribuciones a nivel de estados, introduciendo restricciones en el proceso que sólo permiten la agrupación para aquellas cuyo alófono central es el mismo y pertenecen al mismo estado (inicial, central o final, por ejemplo). Para las distribuciones vacías se optó por usar las correspondientes al modelo independiente del contexto y en el caso de los modelos de silencio no se han entrenado de forma contextual al no haber diferencias significativas en los resultados obtenidos al hacerlo.

El algoritmo de agrupamiento está basado en un cálculo de distancia a partir de la entropía, de la siguiente manera:

$$d(a,b) = N_{ab}H_{ab} - N_aH_a - N_bH_b$$

donde N_a y N_b son el número de ejemplos de las distribuciones a y b disponibles en la base de datos de entrenamiento, respectivamente; N_{ab} es la suma de N_a y N_b y H es la entropía de la distribución correspondiente (en el caso de H_{ab}, ésta se calcula sumando las aportaciones de las matrices de ambas distribuciones):

$$H = -\sum_{c=1}^{NC} p_c log(p_c)$$

donde NC es el número de *codebooks*. La agrupación de distribuciones se realiza sobre aquellas que producen el menor incremento de entropía al ser unidas.

La decisión final acerca del número de distribuciones a usar dependerá de su comportamiento y de la cantidad de datos de entrenamiento, obviamente, remitiendo al lector a referencias como [Gaviña00] para conseguir más detalles al respecto.

5.1.5 Selección automática de unidades independientes del contexto

Además de las alternativas de modelado vistas más arriba, también se planteó en esta tesis el estudio del uso de conjuntos de modelos independientes del contexto generados automáticamente, en lugar de fijar *a priori* la composición de los mismos.

El interés de este enfoque es obvio: si conseguimos un mecanismo de estimación automática del conjunto óptimo de modelos, dada una base de datos de características acústicas determinadas, no tendremos que preocuparnos de la aportación de conocimiento experto para decidir dicho inventario. Como ventaja adicional, si el procedimiento es correcto y los criterios son objetivos (acústicamente hablando), cabe destacar la certeza de que los modelos obtenidos estarán más ajustados al problema acústico con el que nos enfrentamos.

La estrategia usada parte del entrenamiento del conjunto más amplio de unidades con el que contemos, a partir de las cuales se aplica un algoritmo de agrupamiento automático basado en una medida de distancia entrópica. La idea es similar a la usada en el apartado anterior para agrupar

distribuciones de modelos dependientes del contexto, salvo que, ahora sí, las agrupaciones se hacen a nivel de modelo completo, sin restricciones. En este caso, ese enfoque tiene más sentido ya que no se trata de modelos contextuales, siendo nuestro objetivo la reducción del número de unidades globales a considerar, teniendo en cuenta consideraciones de coste computacional y de captura de información acústica.

Los detalles algorítmicos completos pueden encontrarse en [Macías96i], y un ejemplo concreto de las agrupaciones obtenidas se describe en el Anexo D.3, a partir de la página 212.

5.2 Experimentos de selección de unidades y modelado

5.2.1 Estrategia de comparación

Al tratarse en muchos casos de sistemas pensados para ser usados en arquitecturas de hipótesis-verificación, no es razonable recurrir únicamente a la típica comparación de tasa de error para el primer candidato, ya que la diferencia observada para el primer candidato no es tan representativa del comportamiento real del sistema en sistemas con poca resolución acústica (alta tasa de error). Así, en casos como estos, nuestra propuesta es ofrecer, además de la diferencia relativa para este primer candidato, la que corresponde al valor medio de la misma para un cierto rango de la curva de tasa de inclusión.

Puede pensarse que esta media atenuará las diferencias de forma apreciable y que la solución ideal sería comparar directamente las curvas. Estamos de acuerdo en ello, pero ese enfoque es poco práctico ya que la comparación de curvas es raramente sencilla, al no ser las diferencias siempre consistentes para todo el rango considerado, pudiendo producirse cruces entre las que corresponden a sistemas/estrategias distintas. En estos casos está justificado estudiar el comportamiento medio. Igualmente hay que hacer notar que para posiciones elevadas de inclusión, las diferencias observadas se deben a muy pocos ficheros, con lo que se producen grandes saltos, muy bruscos, en las curvas de reducción relativa de tasa.

En la misma línea que el comentario anterior, hay que hacer referencia a la validez estadística de las diferencias observadas: las bandas de fiabilidad tienen que ser calculadas para todo el rango de curva de interés, observándose, obviamente, cómo dichas bandas se solapan a partir de una cierta posición. Esto podría hacernos pensar que hay ocasiones en las que no merece la pena plantear modelos más complejos si se van a usar tamaños de lista para los que hay solape de bandas. En este caso, entendemos que aplica el criterio de *consistencia*[1], más que el estricto de bandas, asumiendo que las mejoras sí son relevantes y que su validez estadística quedaría demostrada si contáramos con tamaños mayores de bases de datos.

Así, en nuestro caso nos centraremos en el análisis, a lo sumo, de cuatro valores: las diferencias para el primer candidato y los valores medios para tamaños de lista igual al 1%, 5% y 10% del tamaño de vocabulario, así como la medida para todo el rango disponible (en forma de curva) o un subconjunto del mismo.

En algunos casos usaremos también la propuesta de evaluación de la disminución relativa de error en función del error del sistema base, como se argumenta en el Apartado 3.5.1 a partir de la página 71 y, por supuesto, las típicas curvas de tasa de error de inclusión en función de la longitud de la lista de preselección, medida como porcentaje sobre el tamaño del diccionario usado.

1. Comentado en el encuadre científico-tecnológico, nos referimos a que si bien las diferencias no son estadísticamente significativas con los márgenes de fiabilidad requeridos, dichas diferencias son consistentes y se producen de forma sistemática.

5.2.2 Modelado discreto y semicontinuo independiente del contexto

La primera observación esperable, es que, en todas las tareas analizadas, para todos los alfabetos, para todos los sistemas y para todos los diccionarios, los modelos semicontinuos superan a a los discretos con amplias diferencias, incluso en las de POLYGLOT, en las que la base de datos de entrenamiento es sumamente reducida, siendo de hecho en este caso mayor esta mejora que en VESTEL, tanto en los sistemas integrados como en los no integrados, llegando a casi un 50% de reducción de error, en ambos casos, para el primer candidato y alcanzando valores similares de reducción media para longitudes de un 1%, 5% y 10% del tamaño del diccionario, como aparece en la Tabla 5-1 donde se indica, entre paréntesis, el rango de tasas de error correspondientes a las posiciones de la curva de error de inclusión consideradas, usando el modelado discreto, para tenerlas como referencia.

Tabla 5-1: Cuadro comparativo de mejora media al incluir modelado semicontinuo para la base de datos POLYGLOT, alfabeto `alf23` y diccionario de 2000 palabras.

Posición de la curva de error de inclusión	**Mejora relativa media para el rango considerado**	
	Sistema no integrado (rango de tasa de error base)	**Sistema integrado (rango de tasa error base)**
1er candidato	49’14% (22’59%)	47’32% (15’21%)
0-1% lista	62’17% (22’59%-2’99%)	56’42% (15’21%-2’14%)
0-5% lista	67’59% (22’59%-0’71%)	49’63% (15’21%-0,48%)
0-10% lista	65’38% (22’59%-0’29%)	33’57% (15’21%-0’18%)

En la tarea de habla telefónica (VESTEL-L) las mejoras también son importantes, produciendo las reducciones de error mostradas en la Tabla 5-2, que no llegan a las de la tarea de habla limpia.

Tabla 5-2: Cuadro comparativo de mejora media al incluir modelado semicontinuo para la base de datos VESTEL-L, alfabeto `alf45` y diccionario de 10000 palabras.

Posición de la curva de error de inclusión	**Mejora relativa media para el rango considerado**	
	Sistema no integrado (rango de tasa de error base)	**Sistema integrado (rango de tasa de error base)**
1er candidato	11’30% (51’36%)	11,59% (31’99)
0-1% lista	28’17% (51’36%-15’25%)	29,62% (31’99%-5’07%)
0-5% lista	34’09% (51’36%-5’66%)	34,97% (31’99%-1’23%)
0-10% lista	36’35% (51’36%-2’90%)	39,43% (31’99%-0’56%)

En la Figura 5-3 se muestran las curvas de reducción relativa de error, en función del error de inclusión del sistema base considerado, al introducir modelado semicontinuo (*sc*) frente al discreto (*disc*) con el alfabeto `alf23`, para las arquitecturas integradas y no integradas y para la tarea sobre POLYGLOT (gráfica de la izquierda) y VESTEL-L (derecha)[1].

En las dos tareas, la introducción de modelado semicontinuo en la arquitectura no integrada produjo mejoras consistentemente más elevadas que la integrada al usar el alfabeto `alf23.`

1. Los resultados de mejora relativa usando diccionarios mayores en la tarea VESTEL-L son muy similares a los mostrados en la Figura 5-3, correspondientes al diccionario de 1952 palabras.

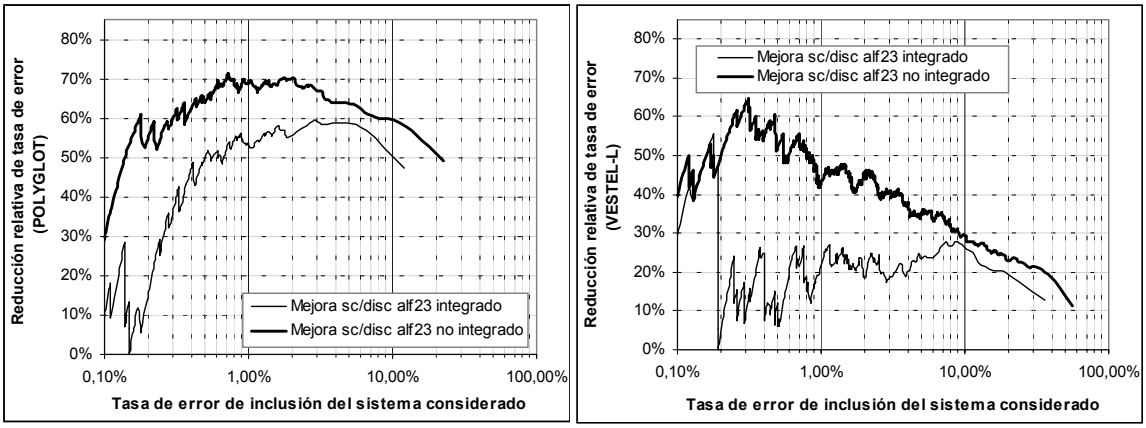

Figura 5-3: Curvas de mejora relativa de error en función del error del sistema considerado al introducir modelado semicontinuo para las tareas POLYGLOT (izquierda) con 2000 palabras de vocabulario y VESTEL-L (derecha) con 1952, para la arquitectura integrada y no integrada usando el alfabeto `alf23`

En la Figura 5-4 se muestran las curvas de reducción relativa de error, en función del error de inclusión del sistema base, al introducir modelado semicontinuo con el alfabeto `alf45`, para las arquitecturas integrada y no integradas y para la tarea sobre POLYGLOT (gráfica de la izquierda) y VESTEL-L (derecha)[1].

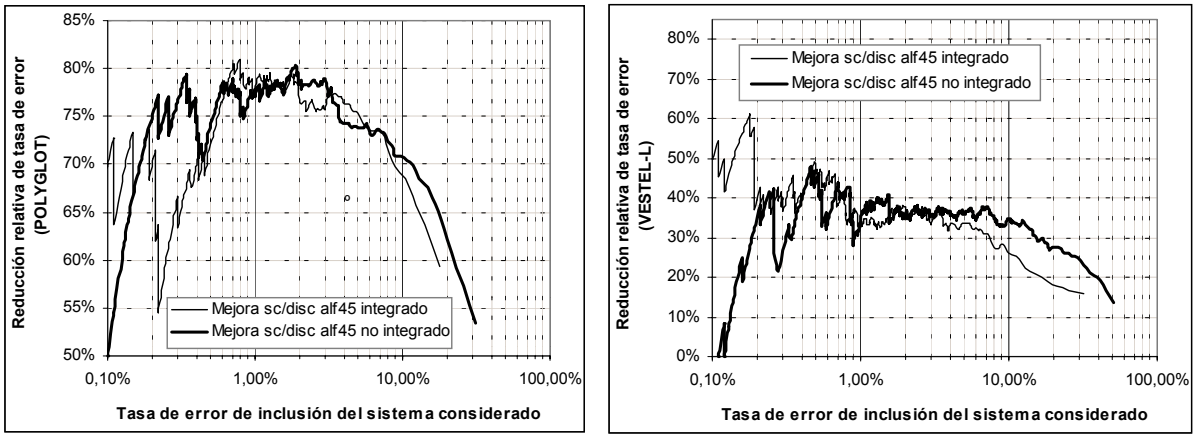

Figura 5-4: Curvas de mejora relativa de error en función del error del sistema base al introducir modelado semicontinuo para las tareas POLYGLOT (izquierda) con 2000 palabras de vocabulario y VESTEL-L (derecha) con 1952, para la arquitectura integrada y no integrada usando el alfabeto `alf45`

En este caso, el comportamiento no es uniforme entre arquitecturas, y no nos ha sido posible encontrar pautas generales de comportamiento, a lo que se suma la falta de datos de entrenamiento necesarios para estimar fiablemente los modelos del alfabeto `alf45` en POLYGLOT.

Se han realizado igualmente estudios detallados de las relaciones entre arquitecturas, potencia de modelado acústico y repertorio de unidades, con las medidas de tasa media de error de inclusión propuestas, para distintas longitudes de lista de preselección, y se ha llegado a la conclusión de que no son fácilmente extraíbles conclusiones de aplicabilidad general, sobre todo en cuanto a las variaciones arquitecturales, salvo algunas pautas como las comentadas y en concreto las referidas al mayor aprovechamiento del modelado semicontinuo en la tarea más sencilla (POLYGLOT).

1. Los resultados de mejora relativa usando diccionarios mayores en la tarea VESTEL-L son muy similares a los mostrados en la Figura 5-3, correspondientes al diccionario de 1952 palabras.

5.2.3 Uso de alfabetos manuales (modelos independientes del contexto)

El estudio de la potencia de modelado de cada uno de los alfabetos considerados se hizo de forma intensiva en la arquitectura no integrada, aunque las conclusiones obtenidas del mismo se han verificado en los sistemas integrados. La dependencia fundamental se encuentra en las características de la base de datos en estudio.

La experiencia previa en nuestro grupo muestra que si la base de datos de entrenamiento es lo suficientemente grande, el incremento del número de unidades a modelar proporciona siempre mejores resultados cuando se aplica a la misma base de datos de entrenamiento [Macías96i], y que los resultados son dispares en la de evaluación.

En POLYGLOT, el alfabeto más simple (`alf23`) ha sido el que ha proporcionado los mejores resultados en las bases de datos de evaluación como se muestra en la gráfica de la izquierda de la Figura 5-5. Sin embargo, al analizar los mismos para la base de datos de entrenamiento observamos que la tendencia comentada en el párrafo anterior no se conserva, produciéndose el mejor comportamiento para el alfabeto `alf33` (como puede verse en la gráfica de la derecha de Figura 5-5). La explicación a este hecho radica en la limitada base de datos de entrenamiento de la que disponíamos.

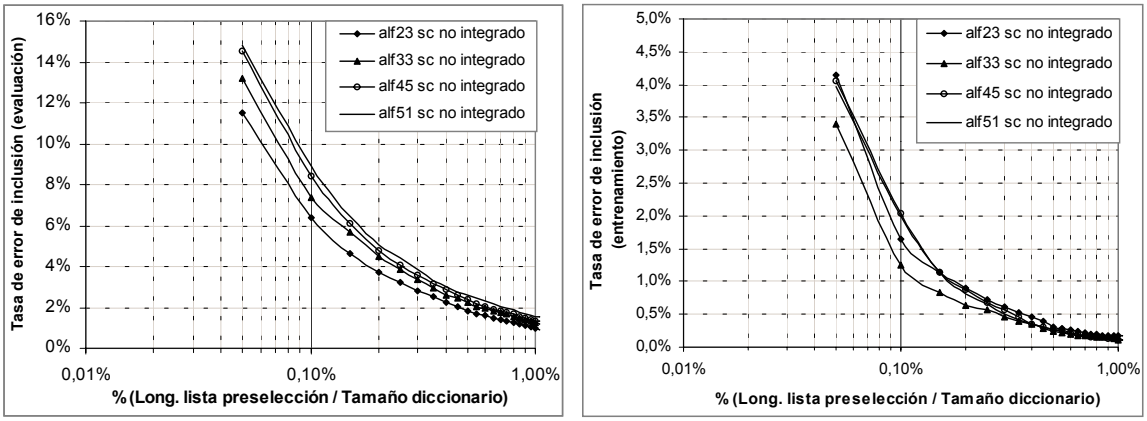

Figura 5-5: Detalle de la curva de tasa de error de inclusión para la tarea POLYGLOT con las bases de datos de evaluación (izquierda) y entrenamiento (derecha), en función del alfabeto manual utilizado

En la Tabla 5-3 se muestra la comparación entre el uso del alfabeto `alf23` y el `alf45`, para la base de datos de evaluación, donde puede apreciarse el considerable deterioro de las tasas obtenidas.

Tabla 5-3: Cuadro comparativo de mejora media al usar el alfabeto `alf45` frente a `alf23` para modelos semicontinuos para la base de datos POLYGLOT y diccionario de 2000 palabras

Posición de la curva de error de inclusión	**Mejora relativa media para el rango considerado**	
	Sistema no integrado (rango de tasa de error base)	**Sistema integrado (rango de tasa de error base)**
1er candidato	-26’63% (22’59%)	-14’94% (15’21%)
0-1% lista	-29’35% (22’59%-2’99%)	-14’88% (15’21%-2’14%)
0-5% lista	-32’09% (22’59%-0’71%)	-1’46%% (15’21%-0,48%)
0-10% lista	-27’93% (22’59%-0’29%)	0’04% (15’21%-0’18%)

En VESTEL-L, sin embargo, se observa un mejor comportamiento en todos los casos a medida que incrementamos el número de unidades (tanto en las bases de datos de entrenamiento como en las de evaluación), salvo cuando llegamos al compuesto por 51 unidades (`alf51`), en las que la tasa de error aumenta ligeramente (aunque las diferencias, si bien consistentes, no son estadísticamente significativas al compararlas con `alf45`). La explicación es el entrenamiento más deficiente de

algunas unidades en `alf51`, como se puede ver en el Anexo D "Alfabetos utilizados", a partir de la página 201, que fue el motivo de la creación de `alf45` a partir de `alf51`. Las mejoras obtenidas entre el alfabeto más simple (`alf23`) y el más complejo de los seleccionados (`alf45`) son consistentes en todo el rango de longitudes de lista y sí son estadísticamente significativas, para un rango razonablemente amplio de valores de la longitud de la lista de preselección (alrededor del 4%). En la Tabla 5-4 se muestran dichos porcentajes de longitud de lista de preselección para los que las diferencias son significativas (entre paréntesis se muestra igualmente la tasa de error correspondiente a esa longitud de lista). La mejora es, en cualquier caso, si no significativa, consistente, por lo que `alf45` será el alfabeto a elegir como óptimo en esta tarea.

Tabla 5-4: Porcentaje de la curva de tasa de error de inclusión para el que las diferencias al usar los alfabetos `alf23` y `alf45` son estadísticamente significativas para la tarea VESTEL-L, con distintos diccionarios y modelado semicontinuo

Diccionario Tarea VESTEL-L	**Longitud de lista[1] para la que las diferencias son estadísticamente significativas (tasa de error en ese punto)**
1952	4’41% (3’96%)
5000-85-15	3’84% (3’98%)
10000-85-15	3’85% (3’92%)

1. Medida como porcentaje del tamaño del vocabulario

En la Tabla 5-5 se muestran a modo de ejemplo los datos cuantitativos de la comparación para el diccionario 10000-85-15 y modelos semicontinuos, siendo estos similares para los otros diccionarios.

Tabla 5-5: Cuadro comparativo de mejora media al usar el alfabeto `alf45` frente al `alf23`, con modelado semicontinuo para la base de datos VESTEL-L y diccionario 10000-85-15.

Posición de la curva de error de inclusión	**Mejora relativa media para el rango considerado**	
	Sistema no integrado (rango de tasa de error base)	**Sistema integrado (rango de tasa de error base)**
1er candidato	7’55% (51’36%)	10’74% (31’99)
0-1% lista	20’13% (51’36%-15’25%)	15’34% (31’99%-5’07%)
0-5% lista	20’17% (51’36%-5’66%)	16’72% (31’99%-1’23%)
0-10% lista	15’24% (51’36%-2’90%)	14’58% (31’99%-0’56%)

En la Figura 5-6 se muestran las curvas de reducción relativa de tasa de error entre el uso de los alfabetos `alf23` y `alf45`, para las arquitecturas integradas y no integradas y la tarea VESTEL-L con el diccionario 10000-85-15. La zona más ruidosa para posiciones entre el 1% y el 10% de longitud de lista, especialmente en el caso de la arquitectura integrada (gráfica de la derecha), se debe al hecho de tener tasas de error muy pequeñas en esos puntos, que hacen que la medida de variación relativa pueda variar de forma más brusca.

Dichas curvas muestran, de nuevo, la mayor capacidad del modelado semicontinuo para aprovechar la información acústica disponible en el entrenamiento, esta vez aplicándola a un alfabeto más completo, aunque en el caso del sistema integrado, las diferencias son menos acusadas en la parte inicial de la curva.

Si nos referimos a la comparación arquitectural, de nuevo los comportamientos son heterogéneos. La introducción del modelado `alf45` fue mejor aprovechada por la arquitectura no integrada para las mejoras promedio hasta el 1%, 5% y 10%. Sin embargo, la mejora para el primer candidato fue consistentemente mejor para la integrada. Si nos fijamos en los comportamientos

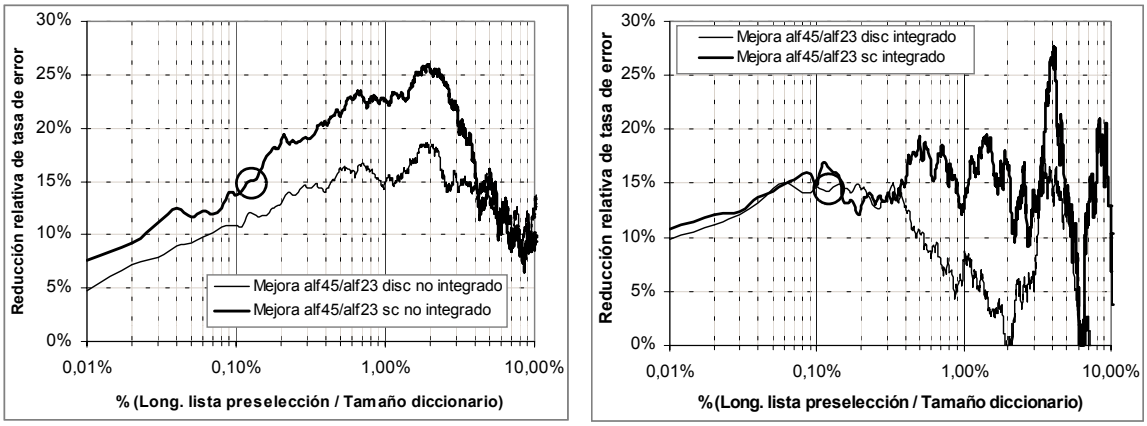

Figura 5-6: Curvas de mejora relativa de error en función del tamaño de lista utilizado, entre los alfabetos `alf23` y `alf45` para modelado discreto y semicontinuo en la tarea VESTEL-L con 10000-85-15, para la arquitectura no integrada (izquierda) y la integrada (derecha)

completos, mostrados en la Figura 5-6, la arquitectura integrada proporciona mayores mejoras hasta una longitud de lista del orden del 0'2% (círculos marcados) y a partir de ese punto, es la no integrada la que se beneficia más del incremento de complejidad del alfabeto. La conclusión es que no es posible establecer pautas de aplicación general en cuanto a la relación entre arquitecturas y la eficacia del uso de distintos alfabetos.

5.2.4 Agrupación automática de modelos independientes del contexto

Partiendo del alfabeto completo (`alf51`), se generaron alfabetos compuestos por agrupaciones de 33 y 23 unidades, y se evaluó la diferencia entre los mismos y sus equivalentes generados manualmente.

Al aplicarlo sobre la tarea telefónica, VESTEL-L, la observación más importante en esta serie de experimentos es que el rendimiento obtenido por los modelos de unidades generadas automáticamente es muy similar a la de los manuales, no siendo las diferencias observadas estadísticamente significativas. Aún más, los resultados obtenidos por los modelos automáticos son ligeramente mejores a los de los manuales, insistimos en que las bandas de fiabilidad se solapan, pero consistentes en un amplio margen de longitudes de lista, para todos los diccionarios y para las arquitecturas integradas y no integradas. En la Figura 5-7 se muestra un detalle de las curvas de error de inclusión para la tarea VESTEL-L con el diccionario de 5000 palabras (5000-85-15) y los alfabetos manual (`alf23`) y automático (`alf_cl23`), ambos compuestos por 23 unidades.

Los mismos experimentos se llevaron a cabo sobre la tarea POLYGLOT, con resultados opuestos: los modelos agrupados de forma automática funcionan significativamente peor que los manuales, como puede verse en la Figura 5-8, incluso enfrentados a la base de datos de entrenamiento.

El problema radica, de nuevo, en el tamaño de las bases de datos disponibles: en el caso de POLYGLOT, los 500 ficheros de entrenamiento por locutor no son suficientes para que las agrupaciones entrópicas realizadas sean realmente consistentes con la tarea acústica. Por el contrario, el considerable tamaño de VESTEL-L produce justamente el efecto contrario.

Nuestra propuesta, es utilizar alfabetos lo más completos posibles si las restricciones de la tarea nos lo permiten, y caso de tener que reducir su número, optar por un agrupamiento automático que nos ofrece la ventaja de estar especialmente adaptado a las condiciones acústicas de nuestra tarea.

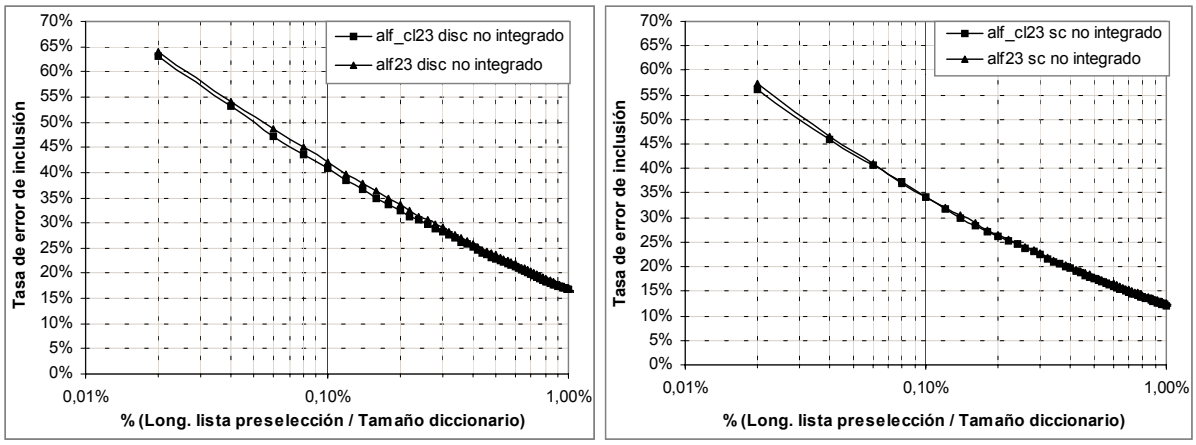

Figura 5-7: Detalle de las curvas de tasa de error para la arquitectura no integrada y alfabetos de 23 unidades: `alf23` (manual) y `alf_cl23` (automático) para la tarea VESTEL-L con un diccionario de 5000 palabras. Modelado discreto (izquierda) y semicontinuo (derecha)

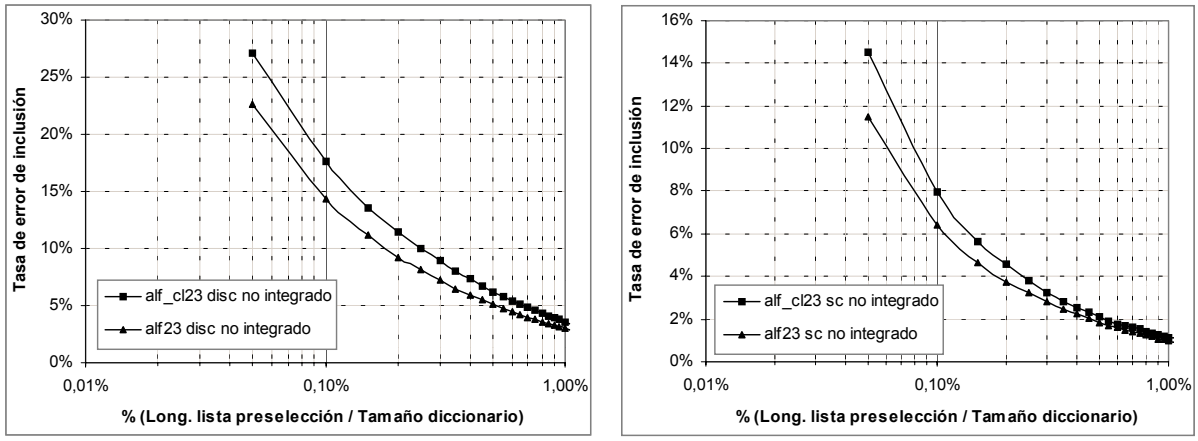

Figura 5-8: Detalle de las curvas de tasa de error para la arquitectura no integrada y alfabetos de 23 unidades: `alf23` (manual) y `alf_cl23` (automático) para la tarea POLYGLOT con un diccionario de 2000 palabras. Modelado discreto (izquierda) y semicontinuo (derecha)

5.2.5 Modelado dependiente del contexto

En lo que respecta al modelado dependiente del contexto, se aplicó únicamente a la tarea VESTEL-L, debido a que los datos disponibles para la de habla limpia eran insuficientes para realizar un entrenamiento mínimamente correcto.

Así, sobre VESTEL-L se realizaron variaciones en el número de distribuciones finales, entre 400 y 2400, con saltos de 200 (valores acotados a partir de los estudios previos descritos en [Gaviña00]). Las diferencias observadas en dichos experimentos no son significativas para un número de distribuciones comprendido entre 600 y 1200. En la Figura 5-9 se muestra la curva de tasa de error de inclusión[1] para dicho rango

Con esos resultados, optamos por usar 800 distribuciones en los experimentos con este tipo de modelado.

1. Tratándose de un sistema integrado con modelos potentes, no tendría sentido usarlo como módulo de hipótesis, estando fundamentalmente interesados en el resultado para el primer candidato, pero se incluye la curva de tasa de inclusión por coherencia con el resto del documento.

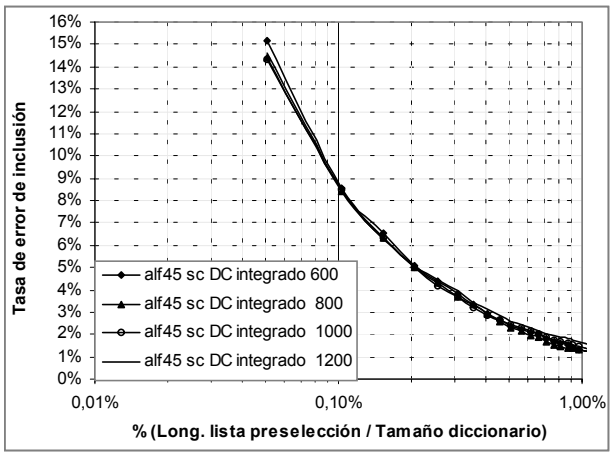

Figura 5-9: Detalle de las curva de tasa de error para la arquitectura integrada con modelos contextuales a partir de `alf45` para la tarea VESTEL-L con un diccionario de 1952 palabras, variando el número de distribuciones usadas.

Por último, en la Figura 5-10 se muestra la reducción de error conseguido al introducir modelado dependiente del contexto en el sistema integrado, en comparación con el uso de modelos independientes del contexto. Como puede observarse, la mejora es muy significativa, estando próxima al 45% para el primer candidato. En dicha gráfica se incluye además la reducción relativa de error al usar el sistema integrado frente al no integrado, con modelado semicontinuo independiente del contexto con el alfabeto `alf45`, consiguiendo igualmente mejoras muy importantes.

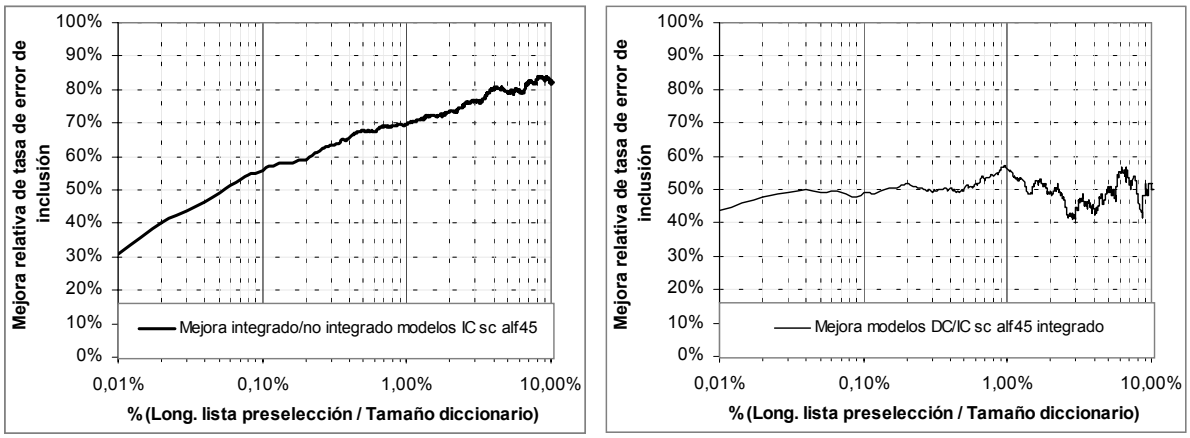

Figura 5-10: Curvas de reducción relativa de tasa de error entre el sistema integrado y no integrado con modelos independientes del contexto. Curva de reducción relativa de la tasa de error entre los modelos dependientes e independientes del contexto en el sistema integrado. Tarea VESTEL-L. Diccionario de 10000 palabras y alfabeto `alf45`.

A la luz de los resultados de la Figura 5-9, todavía no hemos llegado a una estrategia de modelado acústico lo suficientemente potente como para enfrentarse a la tarea VESTEL-L y proporcionar tasas suficientemente altas como para ser viable su aplicación en un entorno real (en el que sería imprescindible una tasa de error por debajo del 5% para el primer candidato). Trabajos en curso en nuestro Grupo están aplicando modelado continuo contextual, con resultados cada vez mejores [Moreno00], pero todo ello pasa por conseguir mayores bases de datos que permitan entrenar de forma robusta dichos modelos.

5.3 Múltiples pronunciaciones

Como se comentó en el encuadre científico tecnológico, nuestra aproximación a la introducción de múltiples pronunciaciones en nuestros sistemas tratará los métodos basados en datos (*data-driven*) y los basados en conocimiento (*knowledge-based*). Igualmente nos ocuparemos de ofrecer visiones alternativas, en la parte de evaluación, a la del simple estudio del impacto en la tasa de error del sistema.

5.3.1 El problema

En la literatura hay multitud de referencias al problema de la introducción de múltiples pronunciaciones (una excelente revisión de la literatura al respecto puede encontrarse en [Strik99]). En general, se trata de abordar el problema de la variabilidad en el modo de producción de la señal de voz debido a diferencias dialectales y modos de articulación específicos de ciertos locutores (variaciones estilísticas, sociales y culturales).

Estrictamente hablando, cualquier sistema de reconocimiento automático de habla se enfrenta de forma *implícita* con variaciones en la pronunciación, dado que los modelos utilizados (HMMs típicamente) se encargan de tener en cuenta todas esas variaciones segmentales y temporales. Sin embargo, lo que se pretende cuando se habla de variaciones de pronunciación, es introducir *explícitamente* conocimiento al respecto de las mismas

A pesar del intenso trabajo en el área, aún no hay soluciones definitivas. La mayor parte de los casos descritos en la literatura se ocupan de estudiar métodos específicos y analizar su impacto en las tasas de reconocimiento.

Cuando se introducen variantes de pronunciación en el léxico (diccionario) de una tarea de reconocimiento, el objetivo es mejorar la precisión de decodificación acústica del reconocedor. Sin embargo, si las variantes introducidas no son adecuadas, la tasa final de error puede aumentar. Con esta restricción, los equipos de investigación son sumamente cuidadosos a la hora de introducir variantes [Adda99]. El problema es precisamente ese: la evaluación planteada típicamente sólo tiene en cuenta el efecto en la tasa global, sin analizar con más detalle la eficiencia de las modificaciones introducidas.

Desde nuestro punto de vista, la investigación realizada hasta el momento no separa claramente la frontera entre el intento de modelar defectos inherentes al reconocedor utilizado y el de modelar fenómenos de variaciones de pronunciación. La diferencia puede parecer sutil, pero a nuestro juicio es fundamental. En [Adda99], por ejemplo, miden la utilidad de las variantes calculando el número de veces en las que se prefiere dicha variante frente a la canónica, argumentando que dicho valor puede indicar la necesidad de variantes o la inadecuación de un único modelo de la palabra, cuando también puede deberse a deficiencias en el repertorio de unidades acústicas con la que se modelan dichas palabras (de hecho, la necesidad de variantes disminuye según se aumenta el número de unidades contextuales usadas).

Nuestra aproximación es, en cualquier caso, eminentemente práctica, en el sentido de abordar la tarea de introducción de múltiples pronunciaciones teniendo en cuenta las limitaciones de la base de datos con la que contamos.

5.3.2 Nuestro enfoque

En nuestro trabajo, sólo nos preocuparemos del nivel segmental, y dentro de él, del referido a variaciones dentro de cada palabra[1]. Del estudio de la literatura y de nuestra experiencia previa, optamos por aplicar tanto la estrategia dirigida por datos (aprovechando la ventaja de contar con

1. Fundamentalmente porque las bases de datos disponibles para esta tesis son de habla aislada. Una referencia relevante a este respecto en nuestro grupo, en el que se consideraba el nivel interno a palabra y también entre palabras es [Ferreiros96]

módulos acústicos muy fácilmente reutilizables para tal fin) como la basada en conocimiento lingüístico explícito (aprovechando el carácter multidisciplinar de nuestro Grupo).

Nuestra idea es apostar por métodos lo más automáticos posibles, no siendo planteables en nuestro caso el abordar técnicas manuales o semi-automáticas, tan costosas en recursos.

El enfoque dirigido por datos es, además, bueno porque nos permite tener en cuenta de forma implícita los problemas de modelado con los que nos enfrentamos[1] y es para nosotros la opción más recomendable. Su problema fundamental es la adecuación del mismo a la base de datos acústica disponible, de modo que los estudios en esta línea no suelen ofrecer resultados generalizables y requieren repetir la metodología cuando cambian las condiciones acústicas (nueva base de datos, nueva región geográfica, etc.).

Las fuentes con las que contamos en el enfoque basado en conocimiento son el conversor grafema-fonema de alta calidad usado en nuestro Grupo, junto con estudios lingüísticos realizados a lo largo de esta tesis acerca de las posibles variantes dialectales o locales del castellano. En cualquier caso, hay que tener especial cuidado en cerciorarse de que los planteamientos basados en conocimiento tienen un reflejo real en la base de datos disponible ya que, de otro modo, podríamos tener problemas de sobre-generación o sub-generación de alternativas, resultando ello en rendimientos peores de los obtenidos hasta el momento.

La estrategia que propondremos en nuestro caso pasa por combinar el enfoque basado en conocimiento con el dirigido por datos, pero no de la forma propuesta en [Strik99], en la que la estrategia basada en conocimiento modela variaciones genéricas y la segunda el resto. Nosotros proponemos una combinación de ambas, siendo el enfoque dirigido por datos, además de aportar información específica, actúa como filtro que valida las propuestas del primero.

5.3.3 Variantes dialectales y variantes culturales

El castellano, como cualquier lengua, presenta una importante suma de variantes dependientes de factores históricos, sociales, culturales, etc. Sin embargo, a la hora de establecer tales variantes y el prestigio o la importancia de cada una de ellas nos encontramos con un problema de fondo: en general, los estudios dialectales o de hablas locales, se basan más en la casuística observada que en estudios porcentuales, y, es habitual que vestigios aislados se consideren de gran importancia, por considerarlos indicativos históricos de la evolución del romance en esa determinada zona. De ahí que, a menudo, se resalten usos no generales, de zonas rurales, o muy restringidos a formas de carácter patrimonial (es decir, incorporadas en el léxico de la zona desde épocas muy tempranas). Por esto mismo, las listas de ejemplos distan mucho de ser exhaustivas, y resulta difícil su generalización [Enríquez00], aunque son casos que habría que considerar.

Otro problema de las descripciones dialectales lingüísticas es que, a menudo, la descripción de un fenómeno se basa en criterios etimológicos, lo cual también complica la definición de una regla de carácter general.

Del estudio realizado en [Enríquez00], se generó un repertorio bastante exhaustivo de todas las variantes diatópicas del castellano (hablas asturianas, leonesas, castellano de los gallegos, catalanes, cántabros, el de las castillas, el extremeño, andaluz, aragonés, etc.) y se elaboró una propuesta ambiciosa de fenómenos razonablemente generales (vocalismos, consonantismos, etc.), de cara a su inclusión en sistemas automáticos de reconocimiento de habla, recogiendo además variaciones y tendencias que surgen de manera espontánea.

En nuestro caso, redujimos aún más dicho repertorio (por ser demasiado productivo en general y con un impacto no siempre relevante), más en la idea de las propuestas de [Ferreiros96] y con un sentido fundamentalmente pragmático.

1. Y volvemos a hacer referencia aquí a la difícil distinción entre modelado de múltiples pronunciaciones y modelado de *defectos* de los reconocedores utilizados.

Más difícil todavía es obtener información cuantitativa del alcance de los efectos considerados. Para ello se necesitaría hacer un etiquetado mucho más fino de las bases de datos disponibles, lo que, por el momento y hasta donde nosotros sabemos, no ocurre en ningún caso; y además, la utilización de unidades acústicas adaptadas con suficientes ejemplos. La información que se puede encontrar al respecto es, como mucho, la referida a la zona geográfica donde se realizó la captura de voz o, en ocasiones, la proporcionada por el locutor como *variante dialectal* que más se ajusta a sus características. Un estudio en profundidad de este tema implicaría un esfuerzo muy importante.

5.3.4 Evaluación

La evaluación del impacto de la introducción de pronunciaciones alternativas en sistemas de reconocimiento automático de habla se hace tradicionalmente midiendo simplemente la reducción en tasa de error obtenida. En este trabajo proponemos una serie de medidas alternativas que complementan la visión anterior (que en ningún caso debemos perder).

Es imprescindible medir el efecto conjunto de la introducción de variaciones en los sistemas (como mejora en la tasa de error), pero es además imprescindible estudiar el efecto particular que se consigue con cada variante. Se trata de evaluar:

- La mejora (o empeoramiento) efectiva global, conseguida en el conjunto de la base de datos de evaluación
- La mejora efectiva marginal, conseguida sobre el subconjunto de la base de datos susceptible de ser mejorado

El primer valor es significativo en cuanto que responde a un comportamiento global, medio, y no interesa que el mismo empeore, ya que responde al efecto sobre la mayoría de los locutores, aunque sería tolerable una mínima degradación si la misma favorece especialmente casos marginales.

El segundo lo es en cuanto a que permite evaluar si locutores especialmente problemáticos o producciones especialmente problemáticas por su variabilidad o dificultad, tengan posibilidades de ser reconocidos correctamente. Para el cálculo de esta mejora es necesario saber a priori cuál es el subconjunto de la base de datos que puede verse beneficiado y eso no es un asunto trivial, en general. Si usamos la estrategia basada en reglas, es fácil de calcular, ya que tenemos la lista de palabras para las que se han añadido variantes. Si usamos la estrategia dirigida por datos, sólo podemos fijarnos en aquellas palabras para las que el sistema propone una secuencia de unidades distinta a la canónica, lo que sucede en la inmensa mayoría de los casos. Por supuesto, la validez estadística de los resultados obtenidos suele ser limitada en estos casos, debido al pequeño número de producciones afectadas.

Así, con la segunda medida pretendemos evaluar la mejora marginal que producen las variantes introducidas. Dicha medida podría parecer irrelevante por su escaso impacto en la tarea global, pero está plenamente justificada porque permite atender a todo un colectivo de locutores o pronunciaciones (o, insistimos, defectos de modelado de nuestro sistema) que si bien es reducido, también necesita ser objeto de nuestro interés de cara a la mejora del rendimiento, marginal pero importante.

Detallando más ampliamente la propuesta de evaluación, y además de evaluar las tasas de inclusión o reconocimiento pertinentes, presentamos a continuación las medidas a realizar para evaluar la introducción de variaciones de pronunciación en los diccionarios de cualquier sistema[1]. Llamaremos diccionarios A y B a los que vamos a comparar y que, en principio serían los que usan pronunciaciones canónicas y múltiples pronunciaciones, respectivamente. Sin embargo, el proceso se plantea como general, dado que podemos estar interesados en evaluar dos diccionarios con distintas variedades de pronunciación.

1. Como se verá, hacemos énfasis en la evaluación orientada a arquitecturas basadas en el paradigma hipótesis-verificación, dado que se hacen medidas de ganancias en posición de palabra reconocida. Sin embargo, el enfoque es extendible a cualquier arquitectura y sistema, sin más que evaluarlo para un número razonable de candidatos.

1. Posición media en la lista de palabras reconocidas en la que se reconoció correctamente una palabra, de aquellas en las que el A mejora respecto al B
2. Posición media en la lista de palabras reconocidas en la que se reconoció correctamente una palabra, de aquellas en las que el B mejora respecto al A
3. Número de palabras en las que el diccionario A mejora respecto al B.
4. Número de palabras en las que el diccionario B mejora respecto al A.
5. Número de palabras en las que ambos diccionarios se comportan de la misma forma.
6. Ganancia absoluta en número de posiciones en la lista de preselección reducidas para las palabras del apartado 1.
7. Idem para el apartado 2.
8. Ganancia relativa en número de posiciones en la lista de preselección para las palabras del apartado 1.
9. Idem para el apartado 2.
10. Número de palabras en las que se reconoció correctamente usando una pronunciación canónica en el diccionario A.
11. Idem en el diccionario B.
12. Número de palabras en las que se reconoció correctamente usando una pronunciación no canónica en el diccionario A.
13. Idem en el diccionario B.
14. Número de palabras en las que se reconoció correctamente usando una pronunciación canónica y había pronunciaciones alternativas en el diccionario A.
15. Idem en el diccionario B.
16. Número de palabras en las que se reconoció correctamente usando una pronunciación no canónica y había pronunciaciones alternativas en el diccionario A (en el caso en el que también A incorpore pronunciaciones alternativas).
17. Idem en el diccionario B.
18. Número de palabras en las que se produjo mejora debido al uso de pronunciaciones no canónicas para el diccionario A.
19. Idem para el diccionario B.

Cada una de las medidas propuestas proporcionan cierta información sobre el impacto de la introducción de múltiples pronunciaciones. Así por ejemplo, los tres primeros valores (1 a 3) muestran una visión de conjunto de las aportaciones de las múltiples pronunciaciones. Las cuatro siguientes (4 a 7) dan idea de la ganancia absoluta o relativa en posiciones de la lista de inclusión. Las cuatro siguientes (8 a 11) ofrecen el detalle del aprovechamiento de las variantes canónicas y no canónicas de los diccionarios, sin entrar en detalle de la mejora que reportan. Las cuatro siguientes (12 a 15) muestran la preferencia de la tarea por pronunciaciones canónicas o no canónicas y, finalmente, los dos últimos (16 y 17) muestran cuántas de las mejoras producidas se deben al uso de variantes no canónicas.

5.3.5 Consideraciones en el compromiso entre impacto y eficiencia

La última consideración importante al respecto de la evaluación de la introducción de variantes es el impacto que tiene en el tamaño del espacio de búsqueda y, por consiguiente, en el tiempo de proceso. Como suele suceder en estos casos, se trata de encontrar un compromiso entre incremento de coste computacional y mejora en la tasa obtenida por el sistema, en las dos dimensiones citadas en el apartado anterior (impacto global y marginal).

5.3.6 Mecanismos de generación de pronunciaciones alternativas basadas en reglas

La formalización de nuestro enfoque basado en conocimiento parte del uso de un sistema de reglas para generar pronunciaciones estándar (conversor grafema-fonema) y una serie de reglas de reescritura de pronunciaciones que permiten introducir las variaciones decididas como importantes. Sin embargo, dichas reglas son expandidas antes de realizar el proceso de reconocimiento en sí, con lo que no se usan como tales, sino que se incorporan al diccionario de la tarea varias entradas por palabra. Éste es, en nuestra opinión, el mejor enfoque si se dispone de memoria suficiente en el sistema. Por supuesto, hay que tener especial cuidado en la selección de las variedades a considerar y su impacto en el tamaño del espacio de búsqueda acústica, lo cual forma parte del primer paso en la metodología a seguir.

Las reglas que finalmente se planteó utilizar, son las incluidas en la Tabla 5-6:

Tabla 5-6: Repertorio de reglas de variaciones de pronunciación

Nombre regla	Descripción	Ejemplo
bs	**Reducción de grupo culto [bs] a [s]**	-bs- > -s-: absurdo > asurdo
ceceo	**uso de [θ] por [s] (está considerado vulgar)**	somos > zomoh
dfinal	**pérdida de [d] final de palabra o cambio por [T] o [t]**	ciudad > ciudá
equis	**Reducción o cambio de grupo culto [ks] a [s] o [gs]**	examen > esamen
gn	**Cambio de [Gn] por [xn] o [n]**	agnóstico > ajnóstico
hue	**Cambio de *hue* por *güe***	Huelva > Güelva
kT	**Cambio de [kT] por [xT], [gT] o [T]**	acción > ación
kt	**Cambio de [kt] por [Tt], [xt], [gt] o [t]**	activar > ativar
participio	**Eliminación de la [d] intervocálida de la terminación del participio**	comido > comío
pt	**Cambio de [pt] por [bt], [xt], [gt], [t]**	aptitud > atitud
sfinal	**Pérdida de la [s] final de palabra**	estudios > estudio
seseo	**uso de [s] por [θ]**	zapato > sapato

Como puede observarse, se trata de fenómenos lingüísticos que, a priori, son razonablemente generales, sin entrar en exceso en detalles más propios de variedades dialectales menos extendidas.

El segundo proceso a realizar dentro de la metodología propuesta es el estudio del impacto a priori de las reglas planteadas en las tareas que nos ocupan. Dicha medida variará en función de las características del sistema sobre el que se vaya a aplicar[1]. Así en nuestro caso, y para cada una de las reglas, se analizó el incremento en número de palabras y alófonos totales, tanto en valor absoluto como en porcentaje sobre los valores correspondientes al diccionario canónico. En la Figura 5-11 se muestra el incremento porcentual en número de entradas, para distintos diccionarios y bases de datos.

Como puede observarse, de las reglas establecidas *a priori*, hay una serie de ellas que tienen un impacto mínimo en la tarea final al no producir entradas de diccionario adicionales o producirlas en un número tan pequeño que será imposible evaluar su eficacia con una mínima fiabilidad (reglas bs, equis, kT, pt). Nuestra propuesta para ellas sería incorporarlas al repertorio de reglas finales básicas si, como mínimo, generan una variante adicional.

1. Podremos medir incremento en el número de palabras, de nodos del árbol léxico, de alófonos en la búsqueda lineal, etc.

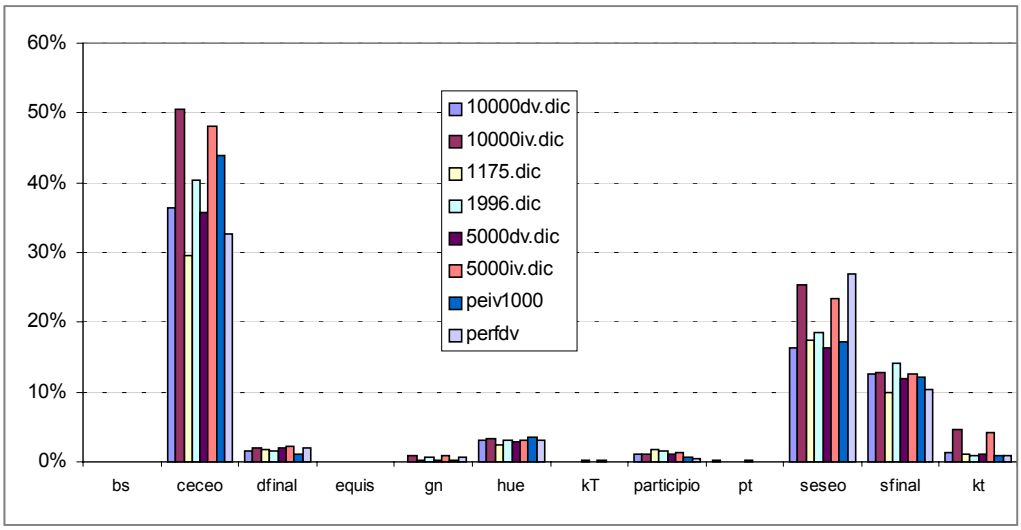

Figura 5-11: Incremento porcentual del número de entradas de diccionario al incorporar las reglas descritas en la Tabla 5-6 para distintos diccionarios y bases de datos

Igualmente, hay un conjunto de reglas cuyo impacto en el incremento del diccionario es muy notable (ceceo, por ejemplo). En estos casos se impone una evaluación objetiva del beneficio producido antes de su incorporación al repertorio final, sobre todo si, como en esos casos, la variación espectral esperada no es lo suficientemente grande como para que nuestros modelos las distingan claramente (se trata de cambio de fricativas).

En el resto de reglas, su impacto es medio o moderado, con lo que es muy probable (con la experiencia que tenemos) que proporcionen mejoras significativas marginales (como se verá en el apartado de experimentación) en casos concretos, aunque la mejora global no sufrirá variaciones importantes.

Los errores adicionales introducidos por las nuevas variantes son de dos tipos:

- Confusiones de palabras existentes por las nuevas, si las mismas están *muy próximas*, acústica y/o léxicamente.
- Confusiones debidas a la introducción de homófonos

El primer tipo de error no es fácilmente predecible, aunque se pueden hacer medidas como las propuestas en Apartado 5.4.1 donde se describen criterios de evaluación de dificultad intrínseca de diccionarios.

El segundo tipo sí que puede tener un efecto claro en la tasa final del sistema, al producir errores sistemáticos. En la Figura 5-12 se muestra el número de homófonos presentes en los diccionarios de 1175 y 1996 palabras de la tarea VESTEL al aplicarle distintas reglas (*bs, ceceo, dfinal, equis, gn, hue, kT, kt, participio, pt, seseo, sfinal*) y una *selección* de ellas (las que aparecen en la Tabla 5-9 en la página 157). En dicha figura puede observarse cómo el incremento de homófonos con respecto al diccionario canónico en la mayor parte de los casos es muy poco importante, salvo para las regla de *ceceo*, *seseo* y *s final eliminada* (además de la *selección*, claro, al incluir varias reglas simultáneamente). Es especialmente interesante notar cómo pueden encontrarse reglas, como la de *hue* o *dfinal*, que generan variantes útiles (como se verá en el apartado de evaluación) y que no incrementan el número de homófonos con respecto a los ya presentes en el diccionario canónico.

Además del estudio a priori mostrado, que se realiza sobre los diccionarios, hay que evaluar igualmente el número de palabras *de las bases de datos* que se verán afectadas.

En la Tabla 5-7 se muestran por ejemplo los homófonos del diccionario canónico y los introducidos por la regla de la *s final eliminada*. Al analizar la base de datos de evaluación PERFDV, aparecen un cierto número de palabras que se pueden ver afectadas (cuyo número aparece entre paréntesis en la Tabla 5-7). El hecho de que haya homófonos no implica directamente un fallo en el

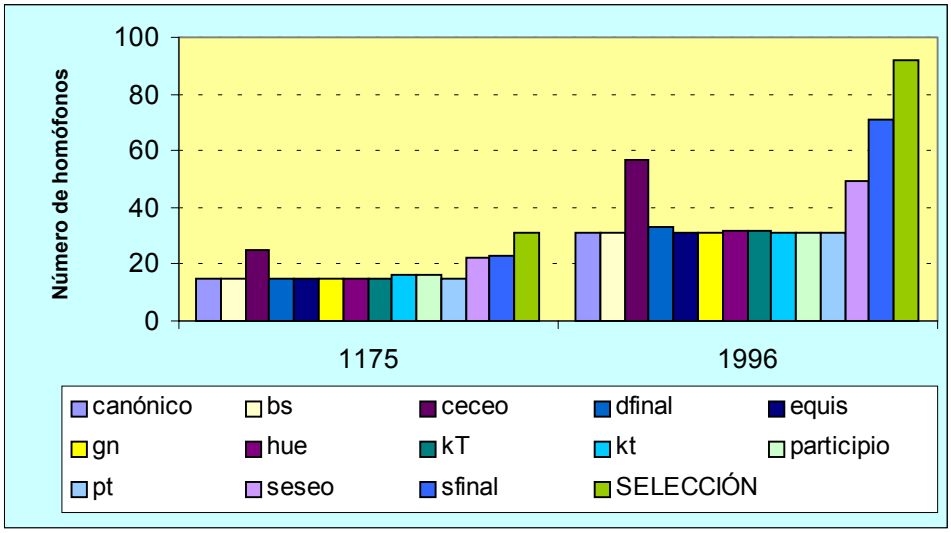

Figura 5-12: Número de homófonos al incorporar algunas de las reglas descritas en la Tabla 5-6 para los diccionarios de 1175 y 1996 palabras (se incluye también el número de homófonos para el diccionario canónico).

reconocimiento, sino que depende de que se contemplen estrategias de desambiguación en el sistema (usando modelos de lenguaje, por ejemplo) y, en caso contrario, de la ordenación que proporcione el módulo acústico (que presentará las palabras reconocidas en una lista ordenadas por probabilidad, pero que a igualdad de probabilidad, las mostrará en un orden indeterminado, no controlable).

En definitiva, es necesario hacer un estudio a fondo del impacto de las reglas aplicadas en el incremento de tamaño del diccionario, en el incremento del número de homófonos y en la base de datos de evaluación.

Sin embargo, las consideraciones hechas hasta aquí se refieren a condiciones de laboratorio, en las que hay un cierto control de la experimentación. En una evaluación en condiciones reales, habría que tener en cuenta que algunas de las medidas de impacto propuestas aquí dejarían de tener sentido, al no poder controlar con tanta precisión las producciones de los usuarios.

Tabla 5-7: Homófonos para el diccionario canónico y algunos de los que introduce adicionalmente la regla de *s final eliminada*

Canónico	**s final eliminada**
balbas balvas	
bernabe bernave	
bolado volado	
cabia cavia	...
chibite chivite	fuente fuentes (3)
enar henar	iglesia iglesias (6)
estebe esteve	mateo (6) mateos (2)
ester esther	riba rivas
hilario ilario	rosa (16) rosas
idolla idoya	tapia tapias (1)
irezabal irezaval	torre (1) torres (10)
llosune yosune	...
maite mayte	
rogelio rojelio	
rut ruth	

5.3.7 Experimentos con variaciones de pronunciación dirigidas por reglas

En esta serie de experimentos, nos centramos en PRNOK5TR, PERFDV y PEIV1000, porque se trata de ver el efecto de variaciones sobre un repertorio de palabras sobre el que tengamos control, lo que nos obliga a usar los diccionarios básicos compuestos por 1175 palabras (para PRNOK5TR y PERFDV) y 1996 (para PEIV1000). El hecho de usar un enfoque basado en conocimiento no tiene los problemas con los que nos enfrentaremos en este sentido en los experimentos dirigidos por datos.

En primer lugar se evaluó el impacto de cada una de las variantes de pronunciación en la tasa global del sistema de preselección, para las listas PRNOK5TR y PERFDV. En ningún caso se encontraron diferencias significativas entre el uso del diccionario canónico y el que incluía las variedades fonológicas utilizadas, como se puede observar en la Figura 5-13, donde se incluye además la curva de mejora relativa de tasa de error. Todo ello a pesar del incremento del número de entradas a considerar, que se indica en la Tabla 5-9.

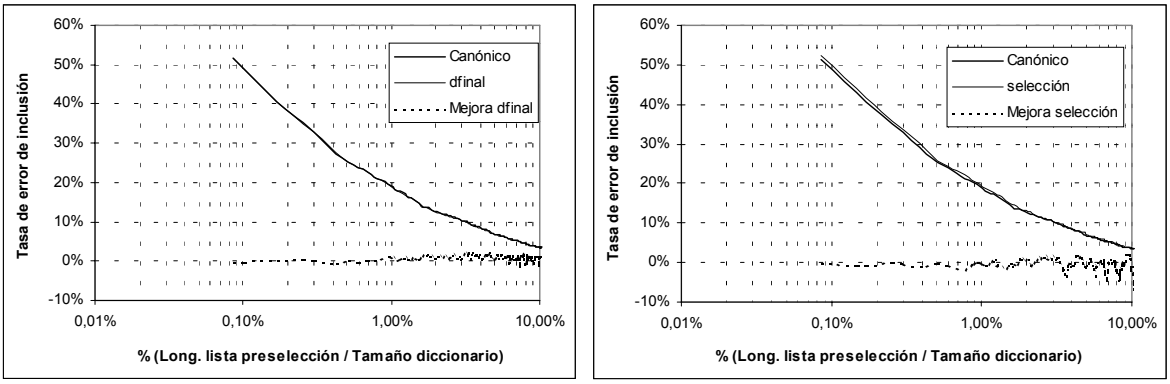

Figura 5-13: Curvas de tasa de error de inclusión para la lista PERFDV usando el diccionario canónico, el resultante de aplicar la regla *dfinal* y el resultante de aplicar la *selección de reglas* de la Tabla 5-9. Se incluye también la mejora relativa de error comparando con el uso del diccionario canónico.

Para tener algún dato cuantitativo, en la Tabla 5-8 se muestran los valores medios de mejora de error al introducir la regla *dfinal* y la *selección* de reglas al compararlo con el diccionario canónico. Como puede verse, los incrementos de error en el caso de la selección son mínimos, todo ello con bandas de fiabilidad solapadas, claro está, y sin un comportamiento consistente.

Tabla 5-8: Cuadro comparativo de mejora relativa media de error al usar el diccionario con la regla dfinal y la selección de reglas frente al canónico. Diccionario de 1175 palabras.

Posición de la curva de error de inclusión	**Mejora relativa media para el rango considerado**	
	Aplicando la regla dfinal (tasa error base, o rango)	**Aplicando la selección de reglas (tasa error base, o rango)**
1er candidato	-0’08% (51’58%)	-0’23% (51’58%)
0-1% lista	0’15% (51’48%-18’7%)	-0’66% (51’48%-18’7%)
0-5% lista	0’89% (51’48%-6’75%)	-0’19% (51’48%-6’75%)
0-10% lista	0’80% (51’48%-3’44%)	-0’35% (51’48%-3’44%)

En la Tabla 5-9 se muestran los valores del número de entradas para cada diccionario al serle aplicadas distintas reglas, así como el conjunto de todas las mostradas en dicha tabla (columna *selección*[1]). Como puede verse, el impacto oscila entre el 1’5% y el 16’98% de incremento para las reglas simples y llega a superar el 30% para la selección combinada.

Tabla 5-9: Número de entradas para cada diccionario en función de las reglas aplicadas (y porcentaje relativo de incremento de tamaño)

		Reglas seleccionadas						efecto de la selección completa
Diccionario		**Canónico**	**dfinal**	**hue**	**participios finales**	**seseo**	**sfinal**	
1175	# palabras	1175	1196	1204	1195	1368	1292	1529
	% incremento	0,00%	1,79%	2,47%	1,70%	16,43%	9,96%	30,13%
1996	# palabras	1996	2026	2059	2025	2335	2277	2675
	% incremento	0,00%	1,50%	3,16%	1,45%	16,98%	14,08%	34,02%

La conclusión es que es muy difícil apreciar el efecto global de la introducción de variedades fonológicas controladas, lo que es debido fundamentalmente a la falta, en la mayoría de los casos, de ejemplos que permitan notar la mejora producida (siendo responsables, como mucho, de mejoras marginales).

Con lo visto, aún nos queda por evaluar el impacto marginal de las variantes introducidas en el sistema, para lo que aplicamos la propuesta de evaluación detallada en el Apartado 5.3.4, que recordamos se centra en evaluar la mejora marginal producida, cuantificarla con más precisión y, por supuesto, dar también información sobre las pérdidas motivadas por el mayor número de entradas de diccionario usadas.

Discutiremos dicha evaluación tomando el ejemplo de la regla *dfinal*, cuyos resultados se muestran en la Tabla 5-10[1] donde se hace referencia al diccionario *canónico* y al resultado de introducir la *variante* de la regla mencionada.

Tabla 5-10: Evaluación cuantitativa del efecto marginal de la introducción de la regla *dfinal* para la lista PERFDV y el diccionario de 1175 palabras

Medida	**Número**	**% del total**	**Valor medio**
Posición media palabras mejores en canónico	120		
Posición media palabras mejores en variante	33		
Palabras mejores con canónico	173	6,91%	
Palabras mejores con variante	16	0,64%	
Palabras iguales	2313	92,45%	
Ganancia Absoluta canónico	238		1,38
Ganancia Absoluta variante	940		58,75
Ganancia Relativa canónico			7,03%
Ganancia Relativa variante			60,15%

1. De todas las reglas mostradas en la Tabla 5-6 de la página 153 hicimos una selección previa teniendo en cuenta su impacto en el incremento porcentual del número de entradas de los diccionarios. Así, dejamos fuera de esta selección las denominadas en la tabla: bs, equis, gn, kT, pt, kt (por su escaso impacto) y ceceo (por el efecto contrario: el incremento excesivo de entradas), quedando las mostradas en la Tabla 5-9.

1. No se han incluido todas las medidas discutidas en el apartado correspondiente por limitar la discusión, si bien todas aquellas podrían ayudar en el proceso de diagnóstico detallado, y de ahí su inclusión en nuestra propuesta.

Tabla 5-10: Evaluación cuantitativa del efecto marginal de la introducción de la regla *dfinal* para la lista PERFDV y el diccionario de 1175 palabras

Medida	Número	% del total	Valor medio
Uso de canónicas en canónico	2502	100,00%	
Uso de canónicas en variante	2487	99,40%	
Uso de no canónicos en canónico	0	0,00%	
Uso de no canónicos en variante	15	0,60%	
Lista de palabras mejores (y número de posiciones que adelantan en la lista reconocida)	(17) david, (6) david, (4) david, (210) david, (29) david, (477) david, (60) david, (51) david, (1) david, (53) david, (1) marian, (24) piedad, (1) ramón, (2) soledad, (3) trinidad		

La primera observación pertinente es la verificación de que el número de palabras *perjudicadas* por la introducción de la variante es muy superior al de las que resultan beneficiadas, 173 frente a 16. Si partimos de esa cifra está claro que el impacto de la aplicación de la regla en el rendimiento global no será nunca apreciable de forma estadísticamente significativa y difícilmente será positivo. El hecho de que mejore se debe fundamentalmente a que las palabras perjudicadas son reconocidas en posiciones más altas de la lista de preselección, y viceversa (en este caso particular, la posición media en la que se reconocieron las palabras perjudicadas fue 120, mientras que para las beneficiadas, de 33).

Sin embargo, si atendemos a la ganancia absoluta en número de posiciones dentro de la lista de palabras reconocidas, observaremos cómo las mejoras producidas por la introducción de la regla (casi 59 posiciones en media) son significativamente mayores que los empeoramientos generados por ella (1’38 posiciones en media). Los valores de ganancia relativa asociados también son muy reveladores: en valor medio, cada palabra perjudicada pierde casi un 7% de posición relativa, pero las beneficiadas ganan un 60’15%.

Por último cabe hacer una referencia a la lista de palabras que han mejorado por la introducción de variantes, en la que puede verse la presencia efectiva de palabras que se han beneficiado de la introducción de la regla sobre la *d* final de palabra (david, piedad, soledad y trinidad), junto a otras como *marian* y *ramón* cuya mejora se debe exclusivamente a la ordenación final que propone el reconocedor.

El mismo estudio se ha llevado a cabo con el resto de reglas descritas anteriormente, con resultados cualitativos similares. A modo de ejemplo final, se incluye en la Tabla 5-11 el resultado de la evaluación propuesta para el diccionario resultante de aplicar la *selección* de reglas descrita en la Tabla 5-9, donde si bien no se alcanzan las diferencias de calidad del ejemplo anterior, sí que se contemplan

Tabla 5-11: Evaluación cuantitativa del efecto marginal de la introducción de la selección de reglas de la Tabla 5-9 para la lista PERFDV y el diccionario de 1175 palabras

Medida	selección	% del total	Valor medio
Posición media palabras mejores en canónico	53,98		
Posición media palabras mejores en variante	26,5		
Palabras mejores con canónico	676	27,02%	
Palabras mejores con variante	79	3,16%	
Palabras iguales	1747	69,82%	
Ganancia Absoluta canónico	3434		5,08
Ganancia Absoluta variante	3053		38,65

Tabla 5-11: Evaluación cuantitativa del efecto marginal de la introducción de la selección de reglas de la Tabla 5-9 para la lista PERFDV y el diccionario de 1175 palabras

Medida	selección	% del total	Valor medio
Ganancia Relativa canónico			22,61%
Ganancia Relativa variante			68,31%

un número considerablemente mayor de efectos de variaciones de pronunciación, también con mejoras marginales muy importantes.

Por último es importante insistir en que, a pesar de que el método seguido está basado en conocimiento, con lo que no estamos sujetos a los problemas de los dirigidos por datos, el impacto estudiado depende fuertemente de las características de la base de datos sobre la que analicemos, en el sentido de que dependemos de que existan producciones de habla que puedan beneficiarse de los cambios introducidos.

5.3.7.1 Aplicación al sistema integrado

Tras verificar las ventajas de la introducción de múltiples pronunciaciones dirigidas por conocimiento en sistemas de preselección, aplicaremos la misma estrategia de incorporación de variaciones fonológicas a la tarea VESTEL-L usando el sistema integrado con modelos semicontinuos dependientes del contexto para el alfabeto `alf45` y 800 distribuciones.

En la Figura 5-14 se muestra la curva de tasa de error de inclusión para la tarea VESTEL-L usando el diccionario canónico y el que contiene la selección de reglas descritas en el apartado anterior en la Tabla 5-9. Como puede verse, se observa un comportamiento ligeramente mejor para el diccionario canónico que para el que incluye variantes (en el primer candidato el empeoramiento relativo es de un 4’37% y un 5’45% para el segundo), pero las diferencias no son significativas estadísticamente, lo que es especialmente relevante, teniendo en cuenta que el número de entradas en el diccionario ha subido de 1952 a 2610, lo que supone un 33’7% de incremento.

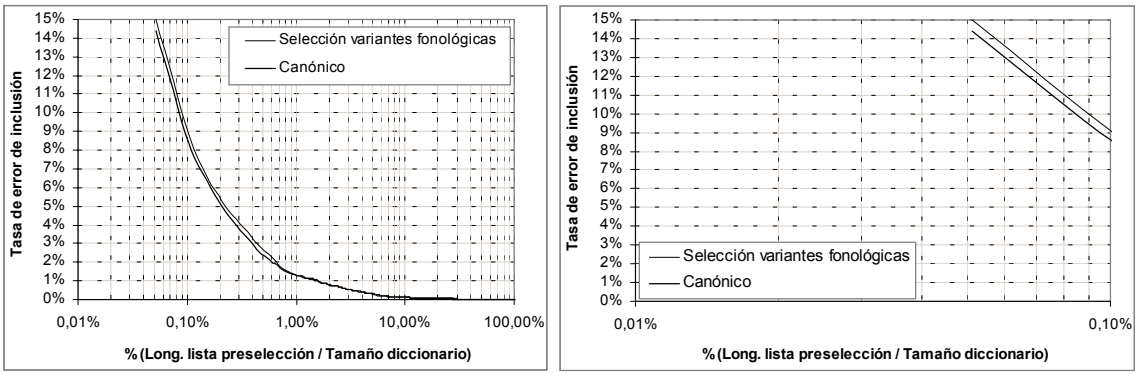

Figura 5-14: Curvas de tasa de error de inclusión para la tarea VESTEL-L usando el diccionario canónico y el que aplica la *selección* de reglas (se muestra la curva completa y un detalle de la misma).

El análisis detallado de mejoras marginales también se ha aplicado en este caso y un ejemplo del mismo lo tenemos en la Tabla 5-12 en la que se puede verificar que los valores obtenidos son similares a los del caso de la arquitectura no integrada. El resto de experimentos asociados a otras reglas muestran tendencias similares, lo que confirma la generalidad de las conclusiones apuntadas más

arriba y valida la estrategia de incorporación de pronunciaciones alternativas basadas en conocimiento, también en sistemas de mayor calidad.

Tabla 5-12: Evaluación cuantitativa del efecto marginal de la introducción de la *selección* de reglas para la tarea LOO completa con el diccionario de 1952 palabras y el sistema integrado con modelos semicontinuos dependientes del contexto

Medida	Número	% del total	Valor medio
Palabras mejores con canónico	548	5’62%	
Palabras mejores con variante	109	1’12%	
Palabras iguales	9099	93’27%	
Ganancia Absoluta canónico	1856		3’4
Ganancia Absoluta variante	1387		12’7
Ganancia Relativa canónico			41’06%
Ganancia Relativa variante			78’56%

Como último comentario, señalaremos que el número de palabras que pierden la primera posición reconocida al introducir variantes de pronunciación es de 114, y las que suben a dicha posición son 53, lo que supone una pérdida neta de 76, menos de un 0’4% de los ficheros disponibles.

5.3.8 Mecanismos de generación de variaciones de pronunciación dirigidas por datos

5.3.8.1 Estrategias de generación

La generación de variaciones de pronunciación se hace en nuestra propuesta a partir de la información proporcionada por un módulo de análisis fonético basado en el algoritmo de un paso que calcula la secuencia óptima de unidades acústicas para cada palabra de entrada. Obviamente dicha secuencia corresponderá en muy pocos casos con la pronunciación canónica de la palabra a reconocer y es precisamente de los errores cometidos por ese decodificador acústico de donde buscamos extraer las múltiples pronunciaciones.

Para nuestros propósitos, veremos el proceso como uno de corrección[1] del diccionario canónico, y estudiaremos distintas estrategias para limitar esa corrección a unos niveles razonables:

- Corrección sin limitación: todas las palabras (cadenas fonéticas) contribuyen a generar variantes
- Corrección limitada a aquellas palabras para las que hay más de un determinado número de ejemplos. Como parámetro de control se incluye el número de repeticiones mínimas necesarias para considerar su inclusión en la lista de nuevas pronunciaciones. La idea es no atender a variantes que no van a poder ser validadas con un mínimo de fiabilidad
- Corrección limitada a aquellas producciones que introducen un número determinado máximo de errores de alineamiento. Como parámetro de control se incluye el número máximo de errores de alineamiento permitidos, en valor absoluto o como porcentaje del número de símbolos de la cadena. La idea es no atender a variantes que introducen una variación excesiva con respecto a la pronunciación canónica.

1. Entendiendo por *corrección* la modificación/incorporación/eliminación de transcripciones del diccionario canónico.

- Corrección limitada a aquellas producciones que producen errores de reconocimiento (refuerzo negativo). Como parámetro de control se incluye el tamaño de la lista de preselección que se considerará como acierto (medido como porcentaje del tamaño del diccionario). La idea es reflejar las variaciones de aquellas palabras que no han sido correctamente reconocidas, con la intención de *recuperarlas*. La crítica fundamental a este enfoque es que permite el aprendizaje de cadenas (variaciones de pronunciación) especialmente malas, lo que puede incidir negativamente en el sistema.
- Corrección limitada a aquellas producciones que producen aciertos de reconocimiento (refuerzo positivo). Como parámetro de control se introduce el mismo que en el caso anterior. La idea aquí es reflejar las variaciones de aquellas palabras que han sido correctamente reconocidas, con la intención de potenciar dicho acierto asumiendo que dichas variaciones ofrecen alternativas reales de pronunciación. La crítica fundamental a este enfoque es que su capacidad de aprendizaje es limitada, al no considerar cadenas problemáticas, lo que puede producir un impacto poco apreciable en el sistema. Comparando el refuerzo negativo y el positivo, podríamos decir que el primero responde a potenciar el aprendizaje de un modelo de error, mientras que el segundo se centra en aprender variaciones de pronunciación.

5.3.8.2 Estrategias de filtrado (reducción)

La idea detrás de las estrategias de filtrado es, en todos los casos, limitar la complejidad introducida en el espacio de búsqueda acústico por el aumento en el número de entradas, dejando aquellas que son realmente relevantes para nuestra tarea, por los beneficios (de nuevo: globales y/o particulares) que reporta en el rendimiento.

Todas las estrategias de filtrado parten de la validación de las propuestas generadas por los mecanismos descritos en el apartado anterior, enfrentando a la base de datos de entrenamiento con los nuevos diccionarios. Nuestra propuesta consiste en estudiar el grado de *uso* de la estructura de árbol usada, entendiendo por grado de *uso* el número de veces en las que cada nodo particular formaba parte del camino óptimo recorrido. Así, nuestro método permite tener una idea muy precisa de hasta qué punto hay alternativas que se utilizan de forma efectiva y cuáles no. El procedimiento práctico consiste en, para toda la base de datos de entrenamiento, alinear cada cadena con el diccionario y anotar el número de veces que se recorre cada nodo.

El tratamiento de las pronunciaciones canónicas presenta varias alternativas. En nuestro caso optamos por analizar el efecto de no favorecerlas de ningún modo o hacerlo (básicamente obligando a realizar un alineamiento con la pronunciación canónica para cada alineamiento de la base de datos de entrenamiento, de cara a mantener un *uso* elevado de las mismas). Los mejores resultados en experimentos previos se obtuvieron con este último enfoque, lo que es comprensible ya que, de otro modo, estaríamos permitiendo que las pronunciaciones canónicas desaparecieran, con el consiguiente perjuicio para las producciones de habla *estándar*.

Una vez disponibles las ocurrencias de cada nodo, analizamos un amplio abanico de métodos de medida de importancia relativa de los mismos, de cara a su eliminación. En este punto introducimos el concepto de *grupo de nodos finales*, entendiéndolo como aquel conjunto de nodos finales que están asociados a una misma palabra (en la estructura de árbol, cada palabra puede tener varias pronunciaciones, lo que se traduce en distintos nodos finales, cada uno asociado a una de ellas). El número de ocurrencias permite estimar valores de probabilidad, que se calculan para cada nodo final. Así, las medidas realizadas fueron las siguientes:

- Impacto en la probabilidad global de cada nodo final (calculado sobre el total de nodos finales): Se eliminan los menos probables.
- Impacto en la probabilidad parcial (calculado sobre el total de nodos del *grupo de nodos finales* al que pertenece el considerado): Se eliminan los menos probables.

- Impacto en la entropía global (calculado como el aumento de entropía que supondría eliminar ese nodo en el conjunto de todos los nodos finales). Se eliminan los que menor aumento de entropía produzcan.
- Impacto en la entropía parcial (calculado como el aumento de entropía que supondría eliminar ese nodo en el conjunto de los nodos de su grupo). Se eliminan los que menor aumento de entropía produzcan.

Insistimos en que en el proceso de cálculo y eliminación, sólo se consideraban los nodos finales, obviamente, dado que usamos una estructura en forma de árbol. En un caso general en el que planteáramos el uso de grafos, habría que considerar también la posibilidad de eliminar nodos en cualquier punto de la estructura, lo que complica notablemente el mecanismo de decisión.

Así, una vez etiquetados convenientemente los nodos, se ordenan de acuerdo con el criterio a seguir en cada caso (de los cuatro vistos) y se elimina un porcentaje determinado de los mismos, con el objetivo de reducir el tamaño del espacio de búsqueda que tenemos tras la corrección y antes del filtrado, lo que especificamos como un porcentaje de incremento con respecto al del diccionario canónico.

La consideración más importante en cuanto a la realización práctica de las medidas es el efecto del tamaño del grupo de nodos finales en las mismas. Para grupos muy pequeños nos encontramos con problemas de estimación y, en general, con posibles valores nulos. Tras una experimentación previa, se llegó a la conclusión de que la mejor forma de evitar dichos problemas era aplicar un suavizado umbral, de la misma forma que describimos para el caso de modelos acústicos.

Por último, mencionar que el cálculo de aumento de entropía presenta problemas prácticos. Si pensamos en la implicación de una eliminación de un nodo, está claro que su pérdida debería implicar el reparto de la probabilidad asociada al mismo entre el resto de posibilidades. La aproximación inmediata al problema es repartir de forma proporcional al resto de probabilidades, pero en ningún momento tendremos la certeza de que dicho reparto se haría de esa forma si volviéramos a realizar el proceso de alineamiento. El cálculo exacto implica un coste computacional sumamente elevado y experimentos previos con listas reducidas mostraron que las diferencias en la calidad de la ordenación no son significativas, si comparamos el método exhaustivo con el aproximado que hemos descrito y que es el finalmente utilizado.

5.3.9 Experimentos de generación de pronunciaciones dirigida por datos

En esta serie de experimentos, nos centramos en PRNOK5TR Y PERFDV, porque se trata de ver el efecto de variaciones sobre un repertorio de palabras comunes en entrenamiento y evaluación, lo que nos obligará a usar el diccionario básico compuesto por 1175 palabras, común a ambas listas. El uso de la lista independiente del vocabulario PEIV1000 no es posible, ya que el diccionario de ésta es totalmente distinto al de las otras dos. Igualmente realizaremos experimentos sobre la misma base de datos telefónica, pero con la división de la misma en segmentos usando la técnica del *leave-one-out*, para ofrecer consideraciones sobre el cambio de condiciones de la tarea y ver cómo afecta a los resultados obtenidos sobre la base de datos comentada anteriormente.

Igualmente, en nuestra comparación usaremos como referencia las curvas de tasa de inclusión, además de valores medios de tasa para un cierto rango de la lista de preselección, completando el escenario donde sea necesario con las medidas propuestas en el desarrollo teórico.

5.3.9.1 Evaluación del proceso de generación

El primer conjunto de experimentos llevados a cabo para evaluar los procesos de generación se orientó a medir el impacto de cada una de las estrategias en la tasa de inclusión del sistema.

A modo de resumen, es importante mencionar que prácticamente todas las estrategias usadas producen mejoras muy importantes sobre la lista de entrenamiento, para todo el rango de variación de

los parámetros de control. Al introducir variantes extraídas del mismo entrenamiento, estamos diciéndole al sistema qué se va a encontrar exactamente a la hora de reconocer sobre dicha base de datos, corrigiendo a la perfección los defectos de modelado de nuestro sistema. La mayor o menor mejora depende de la cantidad de variantes con las que se corrija el diccionario canónico, que serán más o menos en función de la estrategia de generación. En la parte izquierda de la Figura 5-15 se muestran las curvas de tasa de inclusión para la lista de entrenamiento, donde puede observarse que la corrección con la lista completa lleva a una tasa de acierto de prácticamente el 100% desde las primeras posiciones de preselección. Igualmente, el efecto de la generación con refuerzo negativo (en este caso para un valor del parámetro del 1% de la lista) produce una considerable disminución de la tasa de error.

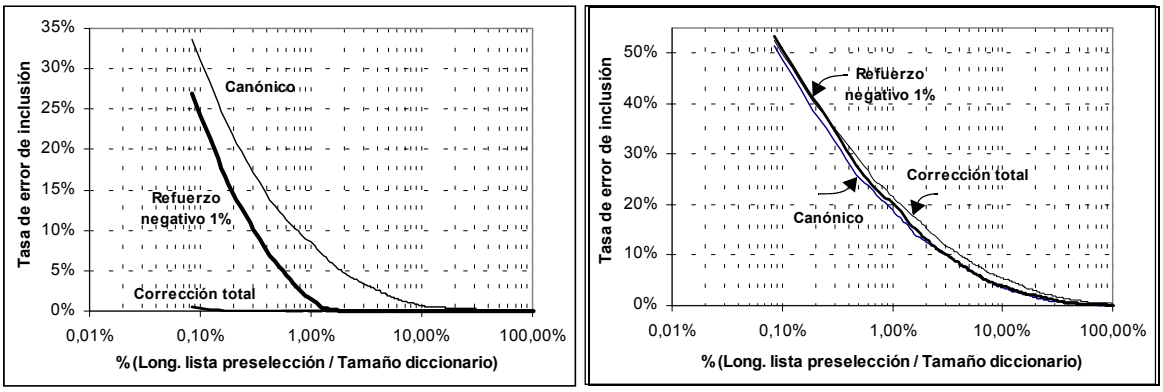

Figura 5-15: Curva de tasa de error de inclusión para las listas PRNOK5TR (izquierda) y PERFDV (derecha), en función del método de generación de variantes utilizado (sin aplicar filtrado).

Sin embargo, el impacto en la lista de reconocimiento es notablemente diferente. En la parte derecha de la Figura 5-15 se puede ver que el rendimiento para el diccionario corregido con la lista de entrenamiento completa es ligeramente menor que para el canónico, pero las diferencias no son tan grandes como podríamos pensar inicialmente, dado que hemos pasado de 5086 nodos del árbol a más de 30000, y de 1175 entradas a 6984 (5809 nuevas). De la misma forma, el comportamiento al usar el diccionario corregido con la estrategia de refuerzo negativo para el 1% de la lista de entrenamiento, obtenemos resultados muy similares a los del diccionario canónico, aunque sigue siendo ligeramente peor, en valor medio. En la Tabla 5-13 se muestran los valores correspondientes a la media de tasa de error de inclusión sobre la base de datos de evaluación, para distintos tamaños de la lista de preselección y distintos métodos de generación, donde puede observarse que los datos son sistemática y consistentemente peores al introducir variaciones, si bien las diferencias no son significativas en todos los casos.

Tabla 5-13: Tasas de error de inclusión medias para la lista PERFDV en función del método de generación de variantes utilizado (sin aplicar reducción)

Media sobre longitud de lista igual a	Tasa de error de inclusión media						
	canónico	Correcc. total	Ref. negat. 1%	Ref. negat. 10%	Ref. posit. 1%	Ref. posit. 10%	MaxErrAbs 3
0% a 1 %	28,11%	30,67%	29,63%	28,43%	30,43%	30,61%	31,29%
0% a 5%	15,35%	17,67%	15,94%	15,53%	17,76%	17,64%	18,21%
0% a 10%	10,54%	12,63%	10,87%	10,68%	12,89%	12,62%	13,06%

Como se ha visto, del estudio del efecto de distintos procesos de generación de variantes de pronunciación se desprende que los resultados son muy distintos para cada uno de ellos al evaluarlas sobre la lista de reconocimiento, pero tras analizar el impacto de las estrategias de filtrado, y como se detalla en el siguiente apartado, se verificó que dicho filtrado es la etapa realmente importante, la que proporciona la verdadera potencia a esta estrategia, con lo que la metodología recomendada parte de

realizar la corrección del diccionario canónico con toda la información acústica disponible, centrando el esfuerzo en la etapa de filtrado de variantes.

5.3.9.2 Evaluación de las estrategias de filtrado (reducción)

Como se comentó en el apartado anterior, a la vista de los resultados obtenidos por los experimentos de generación de variantes para corregir el diccionario canónico, nuestro principal interés se centra en medir el impacto de las técnicas de filtrado de variantes (y reducción del tamaño del diccionario a usar). El primer conjunto de experimentos llevados a cabo para evaluar los procesos de generación se orientó a medir el impacto de cada una de las estrategias en la tasa de inclusión del sistema.

En primer lugar se analizó el efecto de la posibilidad de eliminar las pronunciaciones canónicas, concluyéndose que era contraproducente, como se muestra en la Figura 5-16, para la lista de entrenamiento, donde la pérdida de tasa es muy significativa.

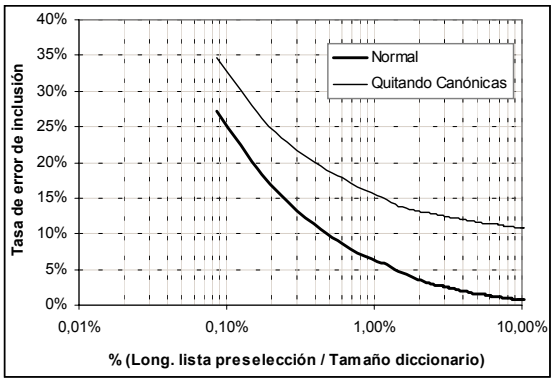

Figura 5-16: Ejemplo de curva de tasa de error de inclusión para la lista PRNOK5TR aplicando reducción y permitiendo (curva *quitando canónicas*) o no (curva *normal*) la eliminación de pronunciaciones canónicas en el proceso.

El segundo factor que se evaluó fue la afirmación hecha en el apartado anterior acerca de la mayor o menor relevancia que tiene la estrategia de generación en la consecución de mejores o peores resultados tras realizar un proceso de filtrado de variantes. Nuestra asunción allí era que lo importante era dicho proceso de filtrado y tal hipótesis se verifica en la práctica. En la Figura 5-17 se muestran las curvas de tasa de inclusión para las listas de entrenamiento (izquierda) y evaluación (derecha) generando variantes con dos métodos (corrección total o sólo con las palabras cuyo número de ocurrencias en la de entrenamiento supera las 5) y filtrando con el método de probabilidad parcial hasta un incremento de diccionario del 250%. Como puede observarse, la elección de uno u otro método de generación influye notablemente en la lista de entrenamiento pero no en la de evaluación, donde las diferencias observadas no son significativas (aunque son ligeramente mejores también para el caso de corregir con todas las variantes disponibles, de forma consistente). Así, guiados por el comportamiento del entrenamiento, usaremos la corrección total como método de generación de variantes.

De cara a obtener un criterio objetivo de evaluación de cada método de filtrado de variantes, analizamos su efecto sobre la base de datos de entrenamiento (PRNOK5TR).

La primera medida realizada fue el valor medio de la tasa de error de inclusión. En la Figura 5-18 se muestran las curvas de tasa media de error de inclusión para un tamaño de lista de un 1% y un 10%, para cada método de reducción de variantes de pronunciación y en función del incremento del tamaño del diccionario permitido en número de entradas (expresado como porcentaje respecto al del diccionario canónico). Como puede observarse, el método que proporciona un peor comportamiento es el de aumento de entropía, en prácticamente todo el rango de valores analizados. La explicación a este efecto es la poca precisión[1] de las medidas de aumento de entropía, sobre todo para grupos de nodos finales pequeños. Para solucionar este problema se intentaron diversas estrategias de modificación del cálculo, sin obtener resultados positivos.

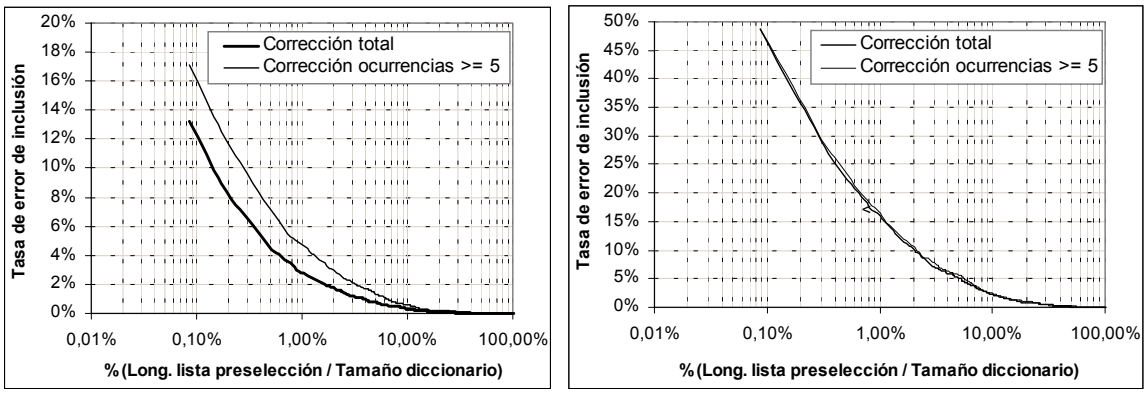

Figura 5-17: Curva de tasa de error de inclusión para las listas PRNOK5TR (izquierda) y PERFDV (derecha) aplicando generación de variantes con dos métodos distintos y filtrado hasta un incremento del 250% del tamaño del diccionario.

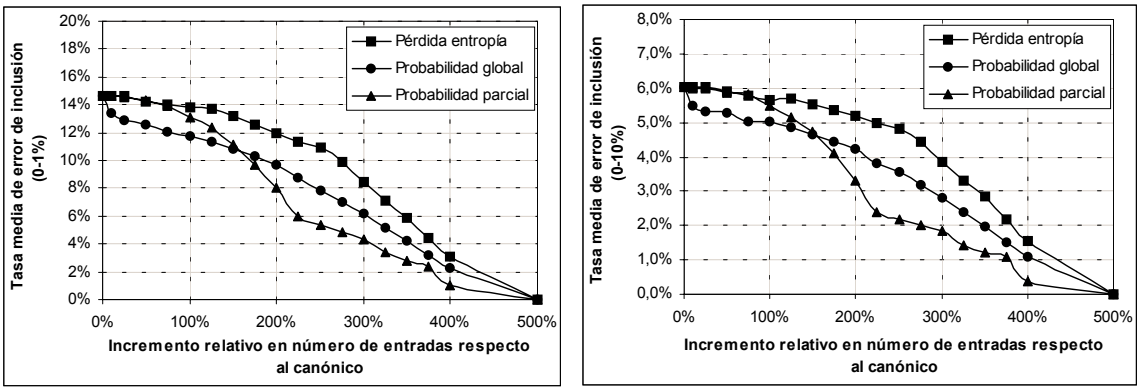

Figura 5-18: Valores de tasa media de error de inclusión para la lista PRNOK5TR aplicando reducción con distintos métodos, en función del incremento relativo en número de entradas realizado sobre el diccionario, para un 1% del número de candidatos (izquierda) y un 10% (derecha).

Así, establecimos que las medidas basadas en medida de probabilidad del uso de variantes, son más potentes que el aumento de entropía, pero, como se puede observar en la Figura 5-18 no está claro cuál de los dos (usando probabilidad global o parcial (la asociada a cada nodo dentro de un grupo)) es mejor.

Para decidir acerca de la bondad de ambos métodos, se optó por evaluar inicialmente la mejora en tasa media de error de inclusión producida al compararlos con el resultado usando el diccionario canónico. En la Figura 5-19 se muestra dicha curva y en ella puede observarse cómo para incrementos reducidos del tamaño del diccionario, el método basado en el cálculo de la probabilidad global funciona mejor, y que dicha tendencia se invierte para tamaños más grandes.

En las gráficas puede apreciarse cómo el método de la probabilidad parcial es el mejor para un amplio rango de valores, a partir de un incremento relativo de tamaño del 150%, y que la máxima diferencia se obtiene en valores alrededor del 250%, que será el valor elegido en nuestro caso.

En la Figura 5-20 se muestra el efecto obtenido al usar los diccionarios reducidos con el método de probabilidad parcial sobre la lista de evaluación PERFDV. Como puede observarse para valores pequeños de incremento del diccionario las variantes de pronunciación introducidas presentan un efecto negativo, pero dicha tendencia se invierte al subir el tamaño, llegando a obtenerse un máximo en 200%, por encima del 150% que nos indicaba el entrenamiento, y próximo a la zona del 250% propuesta anteriormente.

1. No nos referimos a precisión en el cálculo, sino en la existencia de pocos elementos que intervienen en el mismo, lo que puede llevar a grandes errores de estimación.

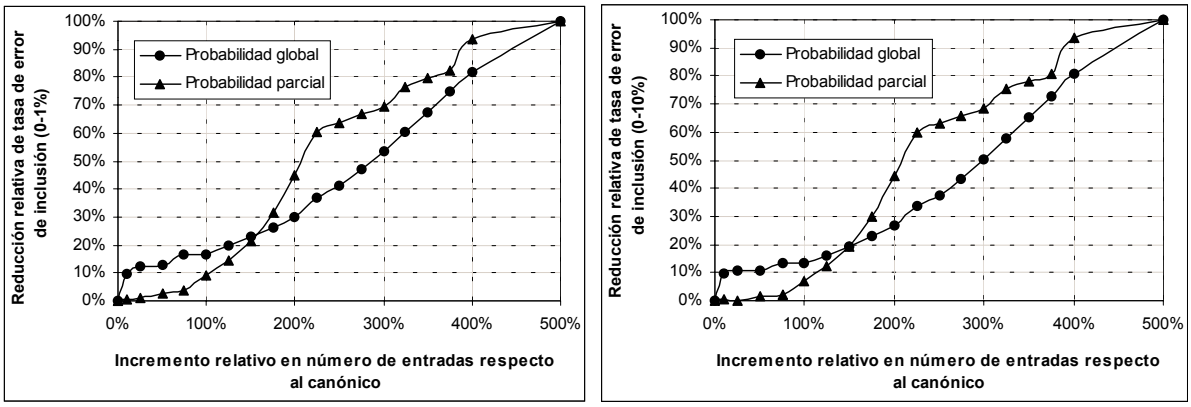

Figura 5-19: Curva de mejora relativa en tasa media de error de inclusión para la lista PRNOK5TR aplicando reducción y comparando con la del uso del diccionario canónico, en función del incremento relativo en número de entradas realizado sobre el diccionario y para un 1% del número de candidatos (izquierda) y un 10% (derecha).

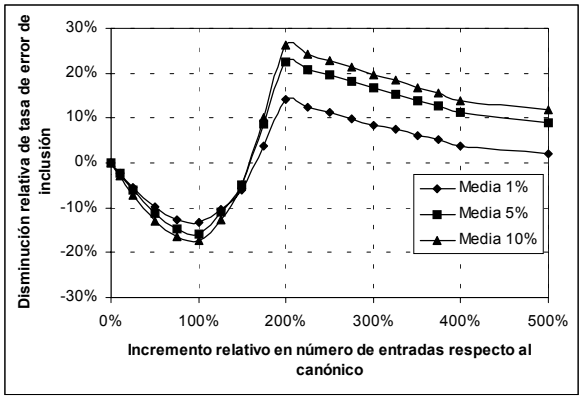

Figura 5-20: Curva de mejora relativa en tasa media de error de inclusión para la lista PERFDV aplicando reducción con el métodos basados en probabilidad parcial, comparando con el uso del diccionario canónico, en función del incremento relativo en número de entradas realizado sobre el diccionario y para un número variable (1%, 5%, 10%) de candidatos

El mismo estudio se hizo para los otros métodos sobre la lista de evaluación y en ningún caso se obtuvieron mejores resultados. El basado en probabilidad parcial captura toda la potencia de decisión necesaria al efectuar decisiones locales dentro de cada grupo de nodos finales. El basado en probabilidad global diluye las aportaciones individuales de cada grupo, suavizando el efecto y disminuyendo la relevancia de las probabilidades particulares.

Merece la pena destacar cómo, a pesar del considerable crecimiento del número de entradas del diccionario, la tasa de inclusión llega a ser incluso mayor que la obtenida con el diccionario canónico, lo que ratifica la bondad de las estrategias de incremento del número de variantes de pronunciación basadas en criterios dirigidos por datos, al modelar deficiencias explícitas de los sistemas sobre los que se aplica.

Para finalizar, en la Figura 5-21 se muestran las curvas de tasa de error de inclusión para la lista de evaluación con el diccionario canónico (etiquetada *antes*) y el corregido (etiquetada *después*) filtrando su tamaño al 250% del diccionario canónico. Se ha verificado la validez estadística de los resultados obtenidos y se ha marcado en la figura una serie de puntos que corresponden a las posiciones en las que las diferencias observadas son estadísticamente significativas. Dado lo limitado de la base de datos (2502 ejemplos), no podemos garantizar la significancia para todo el rango de valores de la longitud de la lista de inclusión, pero sí en la zona de interés, esto es, alrededor del 10% del tamaño del diccionario utilizado. Como dato final, merece la pena destacar que con el diccionario canónico se consigue una tasa de error de inclusión del 2% para el candidato número 208 (17’7% del tamaño de

diccionario) y con el corregido y reducido al 250%, para el candidato número 123 (10’46% del tamaño del diccionario), lo que supone una mejora relativa del 40%

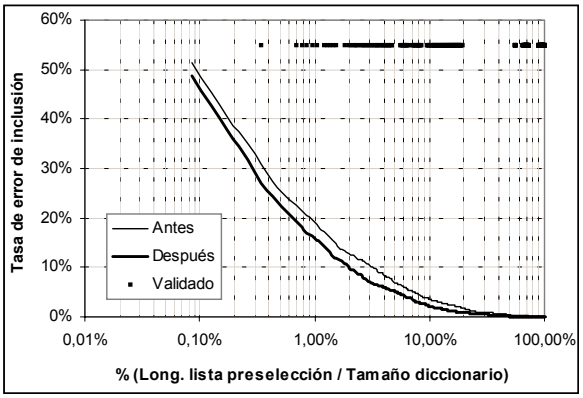

Figura 5-21: Curvas de tasa de error de inclusión para la lista PERFDV, *antes* (usando el diccionario canónico) y *después* de corregir y filtrar (para el diccionario con variantes aplicando reducción al 250% del tamaño del diccionario canónico). Los puntos negros en la parte superior indican aquellos en los que las diferencias son estadísticamente significativas.

La última reflexión a este respecto es volver a insistir en que la aproximación utilizada no puede ser considerada como introducción de múltiples pronunciaciones, sino que, como se ha sugerido, se trata de introducir explícitamente en los diccionarios un modelo de error del sistema en el que intervienen, lo que supone la forma más eficiente posible de incremento de potencia de reconocimiento, (a través del modelado de los errores), fijados los modelos acústicos, claro.

5.3.9.3 Aplicación a la misma tarea (VESTEL) en distintas condiciones (VESTEL-L)

Como se comentó anteriormente, uno de los inconvenientes fundamentales de los métodos dirigidos por datos es la necesidad de repetir la experimentación al cambiar las condiciones de la tarea, la base de datos, los diccionarios, etc.

Así, en este apartado aplicamos las mismas ideas del anterior a la tarea VESTEL-L (que recordamos es idéntica a VESTEL salvo porque se aplicó la técnica del leave-one-out para incrementar la fiabilidad estadística de los resultados obtenidos), usando el diccionario de 1952 palabras: no cambiamos sustancialmente la base de datos, pero sí las condiciones de entrenamiento y evaluación.

Tratándose de una tarea en las mismas condiciones acústicas en la que únicamente repartimos de forma distinta las mismas, podríamos aplicar las conclusiones del apartado anterior, usando el método de corrección con todas las producciones de entrenamiento y filtrando con el método de la probabilidad parcial hasta un tamaño de diccionario igual al 250% del canónico. En la Figura 5-22 se muestran las curvas de tasa de error de inclusión para esta tarea con el diccionario canónico, el corregido sin reducción y el reducido a un 250% y 400% del tamaño del canónico. La primera observación a realizar es el extraordinario comportamiento para el diccionario corregido completo, a pesar del incremento del número de entradas hasta casi 11000. La segunda es la escasa diferencia entre los resultados del canónico y el corregido reducido al 250%. Al contrario de lo visto para PERFDV, las diferencias aquí no son significativas en ningún punto de la curva, aunque en la zona de interés (1%-10%), el corregido reducido a 250% es consistentemente mejor que el otro (consiguiendo una tasa de error del 2% para 166 candidatos frente a los 186 del canónico, lo que supone una mejora relativa del 10’8%). Por su parte, las diferencias entre el diccionario corregido totalmente y el canónico sí son significativas para un amplio margen (mostrado en la Figura 5-22), y lo mismo cabe decir para el reducido al 400%.

La explicación a este efecto parte de un análisis detallado de las condiciones de esta evaluación, en comparación con la vista para PRNOK5TR y PERFDV. En cada prueba parcial (usando *leave-one-out*) usamos una lista de entrenamiento de unas 9000 producciones, y una de evaluación de

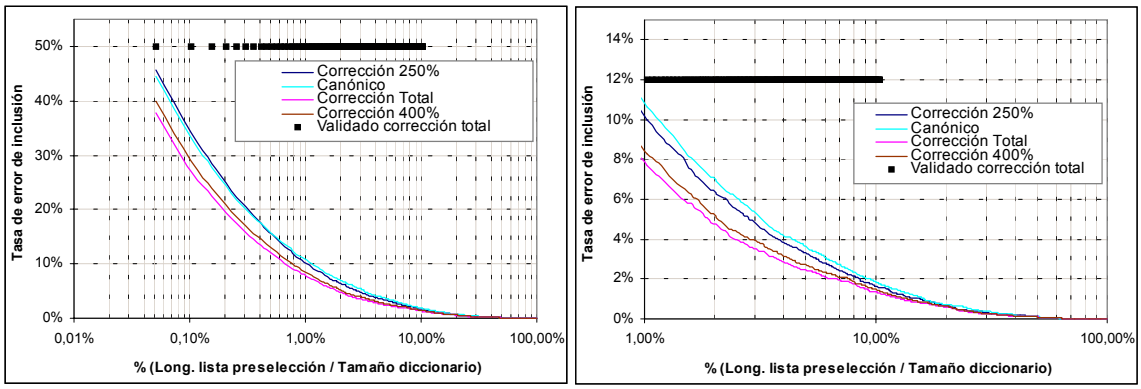

Figura 5-22: Curvas de tasa de error de inclusión para la tarea VESTEL-L usando el diccionario canónico, el corregido totalmente, el corregido y reducido al 250% y 400% del tamaño de aquel y el corregido sin reducción (se muestra la curva completa y un detalle).

casi 1000. En las 9000 producciones de entrenamiento se encuentran ejemplos de casi todas las palabras (unas 1831 en media) del diccionario, pero en las listas de evaluación el número de palabras distintas baja a 478 en media. Con este limitado vocabulario, son pocas las palabras de cada lista de evaluación parcial que van a poder beneficiarse de las variantes introducidas y algunas de las variantes que podrían beneficiar el proceso de reconocimiento son recortadas. Sin embargo, el uso del diccionario corregido completo permite que todas las variantes puedan aportar su efecto en la mejora de los resultados. Ello quiere decir que la longitud de recorte óptima tiene que ser determinada de nuevo para estas condiciones. En la Figura 5-23 se muestra el resultado de dicha evaluación. A diferencia de lo visto para el caso de la lista PERFDV, no se puede establecer claramente la presencia de un punto óptimo, si bien se consiguen mejores resultados que para el diccionario canónico a partir de un incremento del tamaño de un 250%.

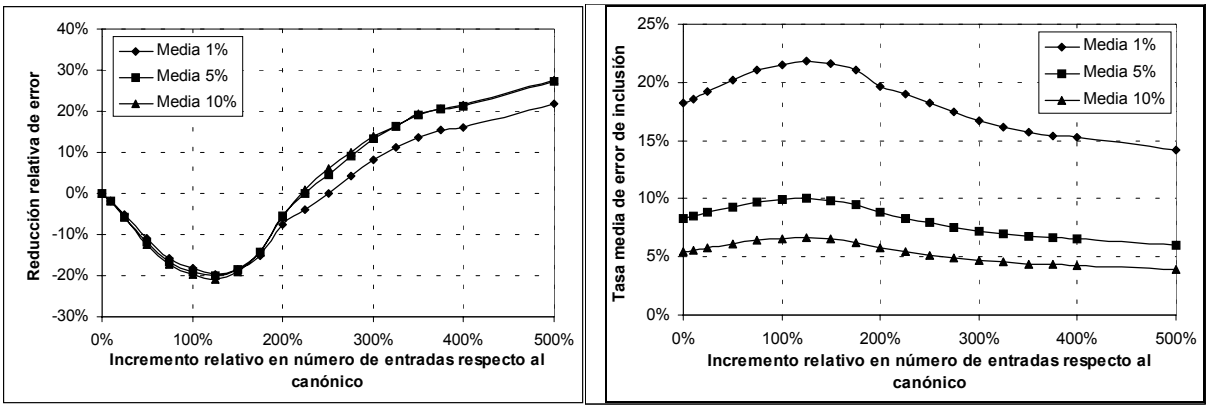

Figura 5-23: Curva de mejora relativa en tasa media de error de inclusión (izquierda) para la tarea VESTEL-L aplicando reducción con el método basado en probabilidad parcial, comparando con el uso del diccionario canónico, en función del incremento relativo en número de entradas realizado sobre el diccionario y para un número variable (1%, 5%, 10%) de candidatos. Curva de tasa media de error de inclusión (derecha).

Así, la conclusión fundamental es que los métodos dirigidos por datos, además de ser muy sensibles a variaciones en dichos datos, también lo son a variaciones en las condiciones de experimentación y que hay que prestar especial cuidado a las mismas para asegurar que el impacto de su introducción puede ser evaluado de forma adecuada, esto es, asegurando que las variantes introducidas van a poder ser utilizadas.

5.3.9.4 Aplicación a un sistema integrado

Tras verificar las ventajas de la introducción de múltiples pronunciaciones en sistemas de preselección, aplicaremos por último la estrategia de corrección y reducción de variantes vista en los apartados anteriores a la tarea VESTEL-L usando el sistema integrado como base y los modelos semicontinuos dependientes del contexto a partir de `alf45` (nuestro sistema más potente).

En la Figura 5-24 se muestran las curvas de tasa de error de inclusión correspondientes al uso del diccionario canónico, y el corregido y aumentado un 400% con respecto a aquél . A diferencia de lo que ocurría al introducir variedades dialectales controladas a partir de reglas, el incremento del diccionario sí tiene aquí un efecto importante en la tasa de error, siendo además dicha pérdida significativa estadísticamente.

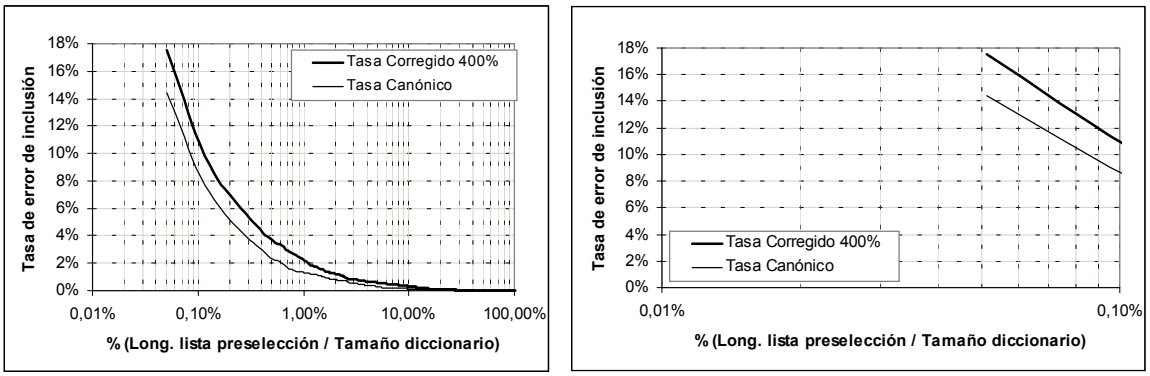

Figura 5-24: Curvas de tasa de error de inclusión para la tarea VESTEL-L usando el diccionario canónico, el corregido y reducido a al 400% del tamaño de aquel y el corregido sin reducción (se muestra la curva completa y un detalle).

En este caso podríamos abordar de nuevo la estimación del incremento óptimo en número de entradas, pero los experimentos realizados al respecto indican que dicho incremento afecta negativamente a los resultados de forma mucho más importante que en el caso de los sistemas no integrados. En la Figura 5-25 se muestra la tasa de error para el primer candidato usando el sistema integrado con modelos semicontinuos dependientes del contexto (800 mezclas) y el alfabeto `alf45`, en función del incremento porcentual de entradas del diccionario canónico, junto con las bandas de fiabilidad asociadas.

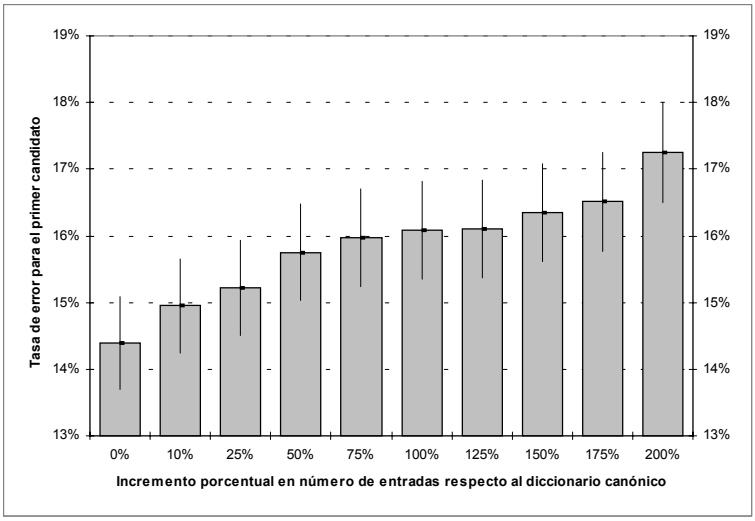

Figura 5-25: Tasa de error para el primer candidato usando el sistema integrado con modelos semicontinuos dependientes del contexto (800 distribuciones) y el alfabeto `alf45`, en función del incremento porcentual de entradas del diccionario canónico.

El hecho de usar un modelado más fino y una búsqueda integrada no se ven beneficiadas en absoluto por el incremento de variaciones de pronunciación. Nuestra explicación a este efecto es la consideración de que dichas variantes han sido generadas con un modelado mucho más pobre, de forma que los errores producidos (y contemplados por tanto en las alternativas de pronunciación) no son coherentes con el nuevo modelado, que produciría otros. Este resultado confirma aún más nuestra observación de que el modelado de múltiples pronunciaciones (sobre todo con métodos dirigidos por datos) atiende fundamentalmente a corregir defectos en la decodificación acústica de los sistemas.

5.3.9.5 Consideraciones de coste computacional

Para acabar este apartado, queremos mencionar que el incremento de coste computacional producido al incrementar el tamaño de los diccionarios del módulo de preselección todavía permiten un amplio margen de maniobra.

En la Figura 5-26[1] incluimos la pérdida relativa de tasa de error obtenida en función de la fracción de tiempo real usado en la tarea de 1952 palabras sobre VESTEL-L usando el diccionario canónico y los que suponen un incremento de un 250% y 400% de entradas en dicho diccionario. Hemos supuesto el peor caso en cuanto a tasa de inclusión, esto es, que los resultados con múltiples pronunciaciones son iguales que los obtenidos sobre el diccionario canónico (hemos visto en apartados anteriores que conseguíamos mejoras con los primeros). Como puede observarse, el aumento de entradas supone un incremento en la demanda computacional del sistema, pero seguimos estando en valores por debajo de tiempo real, dado lo barato del módulo de preselección con respecto al de verificación.

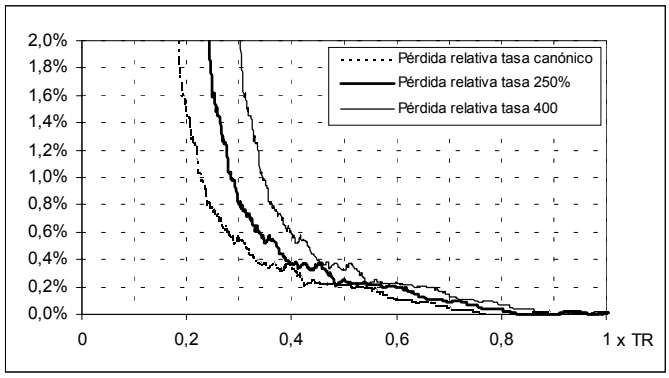

Figura 5-26: Curvas de pérdida relativa de tasa de inclusión para la tarea VESTEL-L en función de la fracción de tiempo real usada, con el diccionario de 1952 palabras en su versión canónica, corregida y reducida al 250%, y corregida y reducida al 400%.

Sin embargo, el impacto en coste computacional para el sistema integrado es mucho más importante, no siendo factible un incremento excesivo del tamaño del mismo, aunque si lo analizamos desde la perspectiva de una arquitectura basada en hipótesis-verificación, podemos plantear el uso del diccionario corregido en la etapa de hipótesis y no modificar los requisitos de la de verificación que podrá trabajar con un número menor de candidatos, con la seguridad de que las tasas obtenidas serán mejores.

5.4 Independencia del vocabulario

Uno de los objetivos de la tesis era diseñar planteamientos que permitieran obtener conclusiones válidas sobre el concepto de *independencia del vocabulario*, intentando aislar aquellos factores que pudieran enmascarar variaciones en la tasa de reconocimiento, no debidas exclusivamente al cambio en el diccionario. El objetivo es validar las posibles comparaciones que pudieran hacerse para

1. Que es similar a la inferior de la Figura 3-5 en la página 61.

evaluar un sistema enfrentado a diccionarios de distintas características, manteniendo o no el número de entradas de los mismos.

La primera observación clara es que la tasa de reconocimiento depende, obviamente, del número de entradas del diccionario utilizado. Sin embargo, al usar medidas de rendimiento como las que aplicamos en esta tesis al hablar de curvas de tasa de error de inclusión, en las que normalizamos el eje de abscisas por la longitud del diccionario utilizado, hace falta abundar más en el estudio de factores relacionados. Así por ejemplo, en la Figura 5-27 se muestran las curvas de error de inclusión para la tarea de reconocimiento de 10000 palabras sobre PRNOK5TR, PERFDV y PEIV1000 . Si tenemos en cuenta que, como se indica en el Anexo B.2 que describe las bases de datos de TIDAISL, PEIV1000 es una base de datos diseñada para hacer experimentación en tareas independientes del vocabulario de entrenamiento (las palabras de las que consta no aparecen en la base de datos de entrenamiento), resulta sorprendente que se comporte mejor que PERFDV, cuyas palabras (grafemas) sí han sido vistas en la lista de entrenamiento. Evidentemente hay factores que influyen en dicho comportamiento y es nuestro objetivo en este capítulo plantear algunas ideas al respecto.

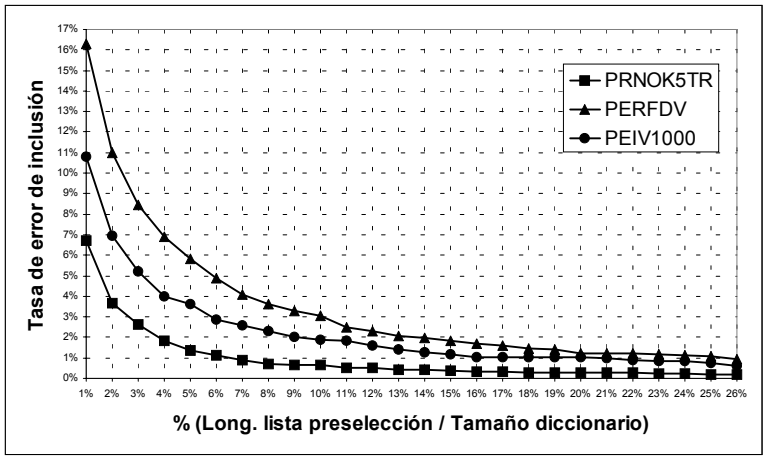

Figura 5-27: Curvas de tasa de error de inclusión para las bases de datos PRNOK5TR, PERFDV y PEIV1000 con los diccionarios de 10000 palabras, en función de la longitud de la lista de preselección (normalizado por el tamaño del diccionario).

Nuestros esfuerzos en este sentido se orientaron en las siguientes líneas de actuación:

- Estudio del impacto en la tasa de reconocimiento del sistema de factores ligados exclusivamente a la composición del diccionario
- Estudio del impacto en la tasa de reconocimiento del sistema de factores ligados exclusivamente a la composición de las bases de datos utilizadas

5.4.1 Criterios de dificultad

De la discusión en otros apartados de esta tesis se estableció la posible correlación entre parámetros relacionados con la longitud de la palabra a reconocer con la tasa de reconocimiento final del sistema. Nuestro punto de partida fue precisamente ese, pero evaluado a partir del conjunto de formas grafémicas asociadas a diccionarios y a listas de bases de datos, calculando las posibles correlaciones entre el rendimiento de un sistema dado y las longitudes medias de aquellas.

Además de esa información, se plantearon parámetros adicionales completando el repertorio a estudiar, basándonos en la idea de que lo que perseguíamos era una medida de la confusabilidad (o dificultad, en definitiva) de cada diccionario o base de datos. Dichas medidas surgen de forma natural a partir de los sistemas estudiados en esta tesis, usando los módulos de acceso léxico (que nos permiten evaluar el impacto de costes de alineamiento entrenados con el soporte acústico real del que disponemos, además de los estándar (Levenstein, por ejemplo)), sin entrar en más consideraciones acústicas ya que nuestro objetivo se centra en no llegar más allá.

Así, el repertorio final de parámetros a estudiar es el siguiente:

- Longitud media de las palabras del diccionario o la lista dada
- Media de la diferencia entre el coste de acceso léxico para la palabra dada y la segunda en la lista de preselección, usando la distancia de Levenstein para los costes de alineamiento
- Media de la diferencia entre el coste de acceso léxico para la palabra dada y la segunda en la lista de preselección, usando los costes entrenados para la tarea considerada
- Combinaciones de las vistas más arriba

Las diferencias de costes de acceso léxico se calculan sobre cada diccionario o lista particular alineando cada entrada del diccionario o la lista con todas las demás del diccionario o la lista. El uso de la distancia de Levenstein nos permite hacer una comparación en la que no interviene la mayor o menor calidad de nuestro modelo acústico, mientras que el uso de costes entrenados aporta la información contenida, entre otras, en las matrices de confusabilidad de unidades elementales de nuestro módulo acústico.

Una alternativa al cálculo propuesto sería el enfrentar las listas con los diccionarios, lo que nos daría una medida más directa de complejidad de la tarea, pero nuestra intención es hacer el estudio de forma independiente.

Como trabajo final, se verán experimentos que analizarán la correlación entre las variables (o combinación de ellas) estudiadas como relevantes de cara a medir la complejidad de un diccionario o lista, y los rendimientos obtenidos en tareas distintas.

5.5 Experimentos sobre complejidad de diccionarios

Los experimentos llevados a cabo en este apartado se realizaron tomando como puntos de evaluación los conseguidos con las listas `loo-tst-?` descritos en el Anexo B.2[1], así como el valor medio para todas ellas. En cada caso se estimaron los parámetros discutidos como relevantes en el Apartado 5.4 y se estudiaron las correlaciones entre los mismos y, en este caso, los valores de tasa de inclusión obtenidos para longitudes de lista iguales a un 1% y un 10% del tamaño del diccionario usado en cada caso. Los diccionarios analizados fueron 1996, 5000-50-50, 5000-85-15, 10000-50-50 y 10000-85-15, lo que nos proporciona un amplio rango de efectos y valores.

5.5.1 Parámetros dependientes de los diccionarios

En la Figura 5-28 se muestra un ejemplo de la tendencia de los valores de tasa de inclusión para una longitud de lista del 1% del tamaño del diccionario, para los 5 diccionarios indicados en el párrafo anterior (sistema no integrado usando modelado semicontinuo independiente del contexto con el alfabeto `alf45`). Como puede observarse, la longitud media del diccionario es un factor relevante para evaluar el rendimiento previsto de una tarea que lo use. La tendencia observada se mantiene también para distintas longitudes de lista (referidas como siempre al tamaño del diccionario).

Es interesante hacer notar que la dependencia vista se mantiene cambiando el diccionario y manteniendo el resto de condiciones de evaluación: misma lista de evaluación, mismo sistema, mismo modelado, mismo número de entradas en el diccionario, etc.. En la Tabla 5-14 se muestra la tasa de inclusión para distintos candidatos evaluando el comportamiento medio para todas las listas `loo-tst-?`, usando diccionarios de 5000 y 10000 palabras con distinta longitud media de palabra. El análisis de validez estadística muestra solape de bandas para puntos contiguos en la gráfica, pero no para los puntos extremos. Además, en este punto enfatizamos lo sistemático de los resultados, coherentemente mejores para todos los casos analizados (para modelos discretos y semicontinuos y

1. Que recordamos eran cada una de las 10 partes en las que dividimos el conjunto de datos de VESTEL para conseguir mayor fiabilidad estadística en los resultados obtenidos.

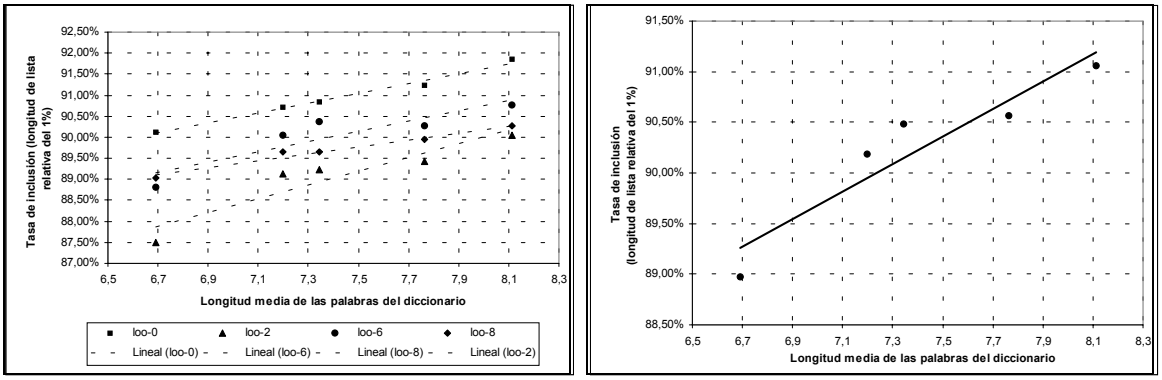

Figura 5-28: Tendencia del efecto de la longitud media de las palabras del diccionario en la tasa de inclusión del sistema para una longitud de lista igual al 1% del tamaño del diccionario. Se incluyen valores para los 5 diccionarios descritos y cuatro de las listas de evaluación utilizadas (izquierda), así como el valor medio para toda la base de datos de evaluación de VESTEL-L

todos los alfabetos disponibles) cuando se usaba un diccionario con el mismo número de entradas y mayor longitud media de palabra. Esto junto con la tendencia observada en la Figura 5-28 da una clara idea de la necesidad de establecer diccionarios homogéneos, al menos en cuanto a longitud media de palabra si nuestro objetivo es realizar comparaciones entre tareas con dependencia o independencia del vocabulario (o cualquier otra en la que haya un cambio de diccionario implicado).

Tabla 5-14: Tasas de inclusión para la tarea VESTEL-L completa en función del diccionario utilizado

	5000-50-50 (longitud media 7'76)	**5000-85-15 (longitud media 7'2)**	**10000-50-50 (longitud media 8'11)**	**10000-85-15 (longitud media 7'34)**
Candidato 1	40,80%	40,07%	34,52%	33,21%
Candidato 10	71,26%	70,59%	64,93%	63,53%
Candidato 100	90,86%	90,59%	86,76%	85,80%
Candidato 1%	86,10%	85,68%	86,76%	85,80%
Candidato 10%	97,66%	97,45%	97,82%	97,53%

En la Figura 5-29 se muestra el efecto de la longitud media de las palabras del diccionario en la tasa media de inclusión para un tamaño de lista del 0'5% del tamaño del diccionario, usando el sistema integrado con modelos semicontinuos dependientes del contexto, con 800 distribuciones y el alfabeto `alf45`[1]. Como puede verificarse, se repite la tendencia vista para el caso del sistema no integrado, si bien en este caso las bandas de fiabilidad de todos los resultados se solapan.

De los otros parámetros analizados, la diferencia de coste medio con la segunda palabra mejor reconocida usando la distancia de Levenstein no mostró correlación ninguna y lo mismo cabe decir de su versión normalizada por el número de símbolos.

Tampoco se encontró correlación apreciable al utilizar las diferencias medias de costes usando los costes entrenados para la tarea (ni la correspondiente versión normalizada).

Este resultado es sorprendente, ya que esperábamos que esa medida de complejidad, dependiente además de costes que habían visto información de la parte acústica sería mucho más relevante. En el apartado siguiente detallamos algún experimento adicional orientado a ofrecer alguna explicación a este efecto.

1. Nuestra intención era coger un tamaño lo más reducido posible de lista, dado que la evaluación se hace en este caso sobre el sistema integrado. El valor del 0'5% implica 10 candidatos

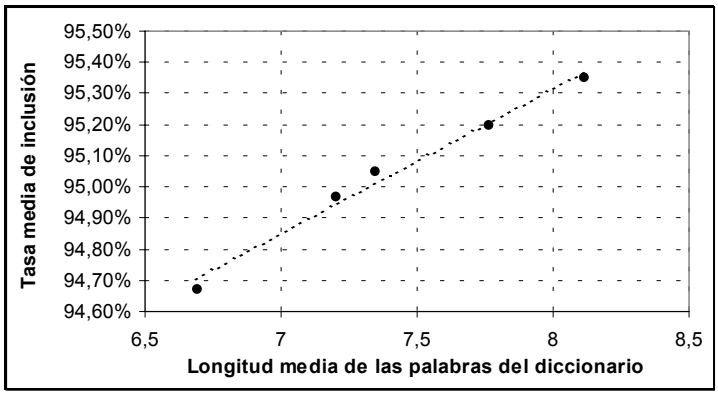

Figura 5-29: Tendencia del efecto de la longitud media del diccionario en la tasa media de inclusión del sistema para una longitud de lista igual al 0'5% del tamaño del diccionario. Evaluado con el sistema integrado sobre la base de datos de evaluación de VESTEL-L.

5.5.2 Parámetros dependientes de las listas (bases de datos usadas)

El mismo análisis que el hecho en el apartado anterior se abordó para las listas de palabras correspondientes a las bases de datos. La dependencia con la longitud media de la lista se muestra en la Figura 5-30 donde, al contrario que en el caso anterior, no hay ninguna correlación apreciable, lo que da idea de la independencia del rendimiento del sistema frente a la longitud de la lista a reconocer.

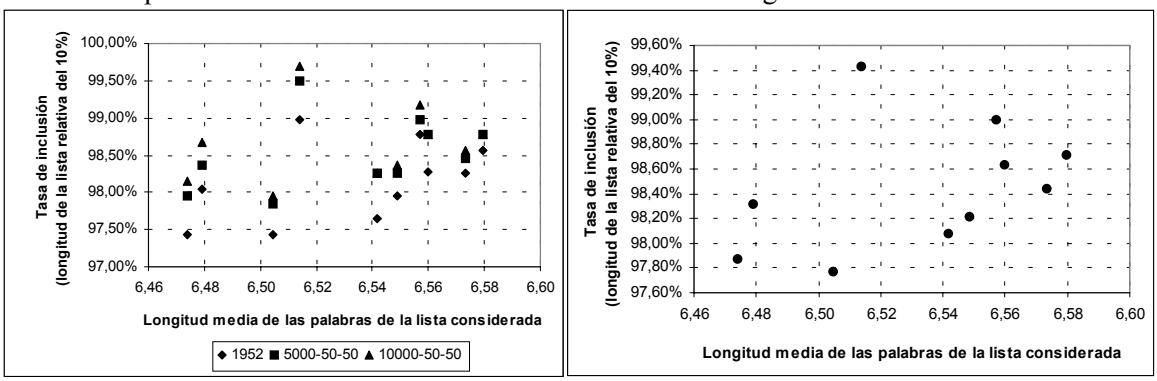

Figura 5-30: Tendencia del efecto de la longitud media de las palabras a reconocer en la tasa de inclusión del sistema para una longitud de lista igual al 10% del tamaño del diccionario. Se incluyen valores para 3 diccionarios y todas las listas `loo-tst-?` (izquierda) y el valor medio (derecha).

Al usar los parámetros dependientes de la distancia de coste medio tampoco encontramos correlaciones, lo que coincide con la observación hecha en el apartado anterior.

En este punto es importante hacer notar que las bases de datos de las que disponemos sólo cubren un vocabulario de unas 2000 palabras, con lo que la artificialidad[1] de la generación de diccionarios usada introduce una dificultad adicional si queremos extrapolar los resultados vistos aquí. Nuestra intuición en el caso de una tarea de la que dispusiéramos todos los ejemplos acústicos de las palabras del diccionario es que el impacto de medidas dependientes de las listas no tendrán una mayor correlación con el resultado esperado que la vista aquí. Basta como muestra mencionar el resultado para el diccionario 1952 en cuanto a falta de correlación, como puede verse en la Figura 5-30, en el que sí se cumple el requisito de disponer de todos los ejemplos acústicos (aunque no en todos los segmentos `loo-tst-?`, sí en el global).

1. En el sentido de que no seremos capaces de verificar el comportamiento de todas las palabras al no disponer de ejemplos acústicos de las mismas.

5.5.3 Parámetros conjuntos

Finalmente se hicieron una serie de experimentos que trataban de evaluar la importancia de considerar simultáneamente parámetros distintos, combinándolos con operaciones simples. Ninguno de estos experimentos ofreció resultados mejores que los observados al usar la longitud media del diccionario utilizado, por lo que es este nuestro parámetro elegido de cara a determinar la adecuación de la comparación de un sistema enfrentado a dos tareas con diccionarios diferentes.

5.6 Conclusiones

En este capítulo se han presentado inicialmente técnicas de selección de unidades de reconocimiento, tanto manuales como automáticas, dependientes e independientes del contexto, y se ha procedido a su evaluación en las tareas planteadas en esta tesis. Se ha mostrado la idoneidad de las técnicas de agrupamiento automático de modelos basadas en medidas de entropía, tanto dependientes como independientes del contexto. En éste último caso se ha verificado que los alfabetos generados automáticamente funcionan igual que los determinados manualmente por un experto, lo que presenta múltiples ventajas, si bien los alfabetos con un mayor número de unidades funcionan mejor si la base de datos de entrenamiento es lo suficientemente grande como para garantizar una estimación adecuada de los modelos. Nuestra propuesta en este sentido es utilizar los alfabetos más completos posibles de los que se disponga, optando por el agrupamiento automático en caso de tener que reducir el número de unidades consideradas.

Se han presentado estudios del impacto de variaciones en el modelado en función de la arquitectura usada y la tarea, con resultados desiguales: en general las relaciones arquitectura-modelado-tarea no son fácilmente interpretables, salvo para algunas tendencias claras, siendo la más destacable la mayor potencia del modelado semicontinuo para tareas acústicamente más sencillas.

En lo que se refiere a múltiples pronunciaciones, se ha hecho un estudio detallado de la problemática de su introducción y, sobre todo, de la evaluación de su impacto, estableciendo la importancia, por una parte, de contar con bases de datos especialmente orientadas a esta tarea y, por otra, de evaluar las mejoras marginales que se puedan producir, proponiendo una serie de métricas que proporcionan información fina sobre este particular.

Se ha abordado el estudio de la introducción de variaciones dialectales dirigidas por reglas, extraídas de un amplio estudio sobre las presentes en las variedades del castellano, con un criterio fundamentalmente pragmático. Se ha estudiado el impacto en los diccionarios (incremento en el número de entradas y homófonos) y se ha evaluado detalladamente siguiendo las métricas propuestas sobre una arquitectura no integrada. La conclusión de este estudio es que las mejoras marginales son muy importantes, manteniendo la degradación de la tasa global en valores mínimos. La aplicación de las mismas ideas a una arquitectura integrada, mucho más potente que la vista anteriormente ha obtenido resultados muy similares, lo que valida nuestra estrategia y método de evaluación.

Igualmente se ha trabajado en generación de variantes dirigidas por datos, introduciendo un nuevo enfoque basado en corrección del diccionario y posterior reducción del espacio de búsqueda resultante, discutiendo sobre distintos criterios de corrección y reducción. Los experimentos han mostrado la potencia de cada criterio, llegando a describir la metodología de trabajo a seguir en esta estrategia. Los resultados obtenidos han sido especialmente buenos, llegando a incrementar notablemente las tasas de inclusión a pesar del considerable incremento del número de variantes introducidas. La aplicación de la misma estrategia y metodología a un sistema integrado ha mostrado un comportamiento contrario: el incremento de variantes ha producido incrementos en la tasa de error, lo que se debe, inicialmente, a que los errores modelados por aquellas son los producidos por un modelado distinto.

A pesar de eso el área de modelado de pronunciaciones está aún en su infancia y el mayor problema con el que se enfrenta es la disponibilidad de bases de datos con un etiquetado específico mucho más fino que el disponen las actuales, lo que queda abierto para trabajos futuros.

También se ha hecho una incursión en el estudio del concepto de dependencia e independencia del vocabulario, lo que nos dio pie a abordar la problemática de la dificultad de diccionarios, proponiendo medidas concretas para evaluar la misma y verificando la sorprendente correlación entre las tasas obtenidas en una tarea con la longitud media de las palabras de los diccionarios dados. La conclusión a este respecto es que los diccionarios deben ser iguales de tamaño medio si queremos comparaciones homogéneas.

6 Conclusiones

En este capítulo incluimos un resumen de las principales conclusiones y aportaciones de la tesis, resumiendo las vistas en cada uno de los capítulos. Se ha incluido además un apartado que recoge las conclusiones sobre el trabajo realizado en la evaluación y otro en el que se indican las principales aportaciones de este trabajo.

6.1 Sobre arquitecturas

A partir del estudio y clasificación de alternativas arquitecturales en el diseño de sistemas de reconocimiento (fundamentalmente enfoques integrados y no integrados, así como los multi-módulo, basados en el paradigma de hipótesis-verificación, como estrategia de diseño de sistemas con el objetivo de reducir la carga computacional necesaria), se ha hecho un estudio teórico de la formulación del comportamiento de arquitecturas multi-módulo, tanto en coste computacional como en tasa de reconocimiento, definiendo una metodología de diseño para determinar la adecuación de módulos particulares de cara a su uso conjunto. Se ha mostrado igualmente lo beneficioso de los reconocedores multi-etapa, que permiten conseguir reducciones muy importantes en tiempo de proceso, manteniendo un rendimiento muy próximo a los máximos alcanzables.

A partir de la evaluación de las arquitecturas implementadas sobre una tarea de habla limpia dependiente del locutor y otra de habla telefónica independiente del locutor, se ha mostrado la importancia del uso de un sistema integrado que permita un guiado explícito en la búsqueda acústica, en las tareas acústicamente más complejas.

Se ha mostrado que es factible conseguir tasas de error de inclusión inferiores al 2% para un tamaño de lista de preselección inferior al 10% del tamaño del diccionario sobre la tarea telefónica con hasta un 85% de ahorro de tiempo de proceso respecto al sistema en un único paso, bajando hasta tan sólo 10 candidatos para la tarea POLYGLOT con un diccionario de 2000 palabras.

6.2 Sobre reducción del espacio de búsqueda

Se ha presentado un estudio y evaluación detallados sobre el impacto del coste computacional de distintas estrategias de organización del espacio de búsqueda orientadas a exploración y búsqueda con algoritmos de programación dinámica: árboles y grafos, deterministas y no deterministas. Se han propuesto igualmente soluciones prometedoras para incrementar la tasa de inclusión obtenible sobre estructuras de grafo (en las que la compresión del espacio de búsqueda produce caídas drásticas de tasa comparadas con la búsqueda lineal o en árbol), combinando de forma óptima parte de la información disponible en el proceso, aunque no se ha llegado a un rendimiento similar al obtenido con los árboles.

La mayor aportación de este apartado ha sido el trabajo sobre estimación de listas variables de preselección, analizando métodos paramétricos y no paramétricos, centrándonos en el uso de redes neuronales como mecanismo estimador. Se ha propuesto una metodología de selección de parámetros de entrada, topologías y métodos de codificación, en base a su potencia discriminativa en una tarea simplificada. Dicha propuesta ha sido ampliamente evaluada y comparada con el enfoque tradicional de uso de listas fijas, basándonos en un mecanismo original de comparación. Se ha mostrado la consistente mejora conseguible con el uso de redes neuronales, aunque no se ha establecido de forma concluyente la fiabilidad estadística de las diferencias apreciadas, dado lo reducido de las bases de datos disponibles.

A partir de los estudios en estimación de longitud de listas de preselección se ha extendido su aplicación, de forma natural, al problema de estimación de fiabilidad de hipótesis, obteniendo buenos resultados en las tareas telefónica y de habla limpia. Se han discutido también los beneficios que proporcionan las redes neuronales frente al uso típico de estimadores directos de fiabilidad, dada su facilidad de uso y su capacidad de integración de múltiples parámetros, proponiéndolas como solución

más adecuada. Finalmente, como trabajo derivado de la estimación de fiabilidad en hipótesis, se ha vuelto a aplicar la idea de estimación de fiabilidad, de forma directa, a la estimación de longitudes de lista, obteniendo excelentes resultados, comparables a los de las estrategias más complejas planteadas en ese capítulo.

6.3 Sobre selección de unidades y diccionarios

Se han presentado y evaluado distintos métodos de selección de unidades de reconocimiento, tanto manuales como automáticas, dependientes e independientes del contexto, mostrando la idoneidad de las técnicas de agrupamiento automático basadas en medidas de entropía si la base de datos es lo suficientemente amplia y representativa de la tarea. Nuestra propuesta final es utilizar los alfabetos más completos posibles de los que se disponga, optando por el agrupamiento automático en caso de tener que reducir el número de unidades consideradas respecto al máximo posible.

Los estudios realizados sobre la relación entre arquitecturas, tipo de modelado y tarea no han mostrado tendencias claras, salvo la demostración de la mayor potencia del modelado semicontinuo para tareas acústicamente más sencillas.

En lo que se refiere a múltiples pronunciaciones, se ha estudiado y evaluado en detalle la problemática de su introducción, estableciendo la importancia de contar con bases de datos especialmente orientadas a esta tarea como único medio de obtener conclusiones relevantes. Las métricas propuestas para la evaluación de mejoras marginales han mostrado ser válidas para nuestros objetivos.

Se han estudiado y evaluado métodos de introducción de variantes de pronunciación basados en conocimiento (dirigidos por reglas, con un criterio fundamentalmente pragmático) y dirigidos por datos, analizando el impacto en los diccionarios y los resultados obtenidos con las métricas propuestas.

En lo que respecta los métodos basados en conocimiento, se ha concluido que las mejoras marginales obtenidas son muy importantes, manteniendo la degradación de la tasa global en valores mínimos, tanto para la arquitectura integrada como la no integrada.

En los métodos dirigidos por datos se ha propuesto un nuevo enfoque basado en la corrección del diccionario y posterior reducción del espacio de búsqueda resultante. En este aspecto se ha descrito la metodología de selección de mecanismos de corrección y filtrado, siendo los resultados obtenidos especialmente buenos para los módulos de preselección, llegando a incrementar notablemente las tasas de inclusión a pesar del considerable incremento del número de variantes introducidas. La aplicación de la misma estrategia y metodología a un sistema integrado ha mostrado peores resultados, lo que se explica en parte por la diferencia de los modelos de error que fueron los aprendidos sobre los modelos acústicos más pobres, quedando para el futuro el estudio detallado de todo esto. Se ha mostrado igualmente cómo los métodos dirigidos por datos no sólo son sensibles a las bases de datos usadas, sino también a la distribución de los conjuntos de entrenamiento y evaluación seleccionados, siendo imprescindible un estudio previo de si va a ser posible medir el impacto de las variaciones introducidas, lo que no siempre sucede.

También se ha hecho una incursión en el estudio del concepto de dependencia e independencia del vocabulario, lo que nos dio pie a abordar la problemática de la dificultad de diccionarios, proponiendo medidas concretas para evaluar la misma y verificando la sorprendente correlación entre las tasas obtenidas en una tarea con la longitud media de las palabras de los diccionarios dados. La conclusión a este respecto es que los diccionarios deben ser iguales de tamaño medio si queremos comparaciones homogéneas.

6.4 Sobre evaluación

Tratamos las propuestas y conclusiones sobre evaluación como apartado independiente ya que ha sido un aspecto especialmente cuidado en el desarrollo de la tesis, habiéndose realizado propuestas en distintos capítulos.

En el apartado de evaluación de sistemas multi-módulo, se han propuesto tres estrategias de medida de rendimiento especialmente orientadas a nuestros intereses, pero que son extensibles a cualquier arquitectura basada en el paradigma hipótesis-verificación e, incluso a arquitecturas en un único paso:

- La primera, consistente en medir las curvas de tasa de error de inclusión en función, no del número de candidatos seleccionados, sino del porcentaje que dicho número implica referido al tamaño del diccionario usado en la tarea, para poder efectuar comparaciones entre tareas con distinto tamaño de vocabulario
- La segunda, una medida alternativa de disminución relativa de tasa de error en función de la tasa de error del sistema base, lo que puede permitir obtener una visión más rica del efecto de distintas estrategias en sistemas con tasas base muy distintas.
- La tercera, en la que se evalúan valores medios de tasa de inclusión para un cierto tamaño de la lista de preselección, lo que permite una comparación más cómoda que la que evalúa la curva completa de tasa de inclusión.

Cada una de ellas nos ha sido de utilidad para evaluar aspectos concretos de interés a lo largo de la tesis.

En el capítulo de estimación de longitudes variables de listas de preselección basadas en redes neuronales se ha propuesto un mecanismo original de evaluación que permite considerar tanto los valores particulares obtenidos como la sensibilidad a variaciones en parámetros del sistema.

En el apartado de múltiples pronunciaciones se ha propuesto también un mecanismo original de evaluación de las mejoras marginales obtenidas, como medida fundamental para decidir sobre la bondad de las variantes introducidas.

El proceso de evaluación de los resultados se ha cuidado especialmente, atendiendo a tareas distintas (habla telefónica y habla limpia), para obtener conclusiones en un amplio espectro de situaciones y matizando los resultados a la luz del análisis de fiabilidad estadística.

6.5 Principales aportaciones

Resumiendo, las principales aportaciones de la tesis son las siguientes:

- Desarrollo de una metodología de diseño de sistemas multi-módulo
- Establecimiento de criterios generales de diseño en cuanto a modelado y arquitectura
- Evaluación de distintas estrategias de organización del espacio de búsqueda, en tiempo y tasa, con una propuesta de incremento de ésta para estructuras compactas de tipo grafo
- Propuesta de un mecanismo novedoso de estimación de listas variables de preselección basado en redes neuronales, extendiéndolo a tareas de estimación de fiabilidad de hipótesis
- Para la selección de unidades de reconocimiento: Propuesta de estrategias dirigidas por datos (agrupación basada en entropía) y basadas en conocimiento (repertorio manual), y propuesta de selección en función de las características de la base de datos de entrenamiento

- Para la selección de diccionarios a través de la incorporación de múltiples pronunciaciones: Propuesta de estrategias dirigidas por datos (con mecanismos específicos de generación y filtrado) y basadas en conocimiento (reglas), y propuesta de combinación como estrategia más eficiente
- Propuesta de novedosas métricas y estrategias de comparación específicas, más adaptadas que las tradicionales a las condiciones de los sistemas en estudio

7 Líneas futuras

Dada la amplitud y variedad de los temas planteados en esta tesis, han surgido multitud de propuestas de continuación.

Al igual que hicimos en el capítulo de conclusiones, organizaremos éste de acuerdo a los grandes apartados tratados en este documento.

7.1 Sobre arquitecturas

Un aspecto relevante que no ha podido ser abordado en este trabajo es la introducción de las técnicas alternativas de entrenamiento descritas en el encuadre científico-tecnológico, como las de entrenamiento correctivo, estimación de máxima información mutua (*MMIE*) y métodos basados en error de clasificación mínimo (*MCE*) y descenso generalizado probabilístico (*GPD*), con el objetivo de mejorar la calidad de los modelos utilizados. En este mismo sentido, habría que hacer un esfuerzo importante para aplicar técnicas de entrenamiento dependiente y conjunto [Chiang96], en aquellos sistemas multi-módulo disponibles, en los que, en general, se hace independientemente.

Tampoco se ha podido explorar la prometedora estrategia de combinación ponderada de costes (*scores*) o probabilidades entre distintos módulos de la cadena de hipótesis-verificación, como estrategia óptima para conseguir tasas previsiblemente mejores de las que serían obtenibles por uno solo, en la línea de propuestas como la usada en el sistema POLYGLOT descrito en [Leandro94].

La propuesta más compleja referida a la formulación desarrollada para sistemas multi-módulo sería la extensión de los planteamientos teóricos realizados a habla continua, lo que es fundamental para acercarnos más a las tareas que más ampliamente se están utilizando en la actualidad. En este caso, habría que evaluar el impacto de medidas relacionadas con la complejidad del grafo de alternativas generado por el módulo de hipótesis (o cualquier estructura usada como entrada al de verificación) en la tasa global del sistema. En esta misma línea sería fundamental la introducción de modelos de lenguaje que no han sido incorporados en las evaluaciones realizadas en esta tesis.

En el apartado algorítmico, queremos mencionar que los costes utilizados en el acceso léxico no son completamente contextuales, con lo que planteamos como estudio futuro el uso de costes contextuales (dependientes de las unidades a la derecha y a la izquierda de la considerada), con lo que, si la base de datos de entrenamiento es suficientemente amplia, es previsible que se incremente el rendimiento del sistema.

7.2 Sobre reducción del espacio de búsqueda

En primer lugar creemos que es fundamental la aplicación de técnicas tradicionales de búsqueda en haz en las arquitecturas discutidas en la tesis, como las descritas en [Ortmans97b][Colás99], combinadas con el resto de métodos propuestos, los de estimación de listas variables de preselección. De esta combinación será posible reducir aún más la demanda computacional y permitir abordar de tareas más complejas en tiempo.

En la sección correspondiente a la búsqueda sobre grafos, se propusieron métodos para mejorar la ordenación producida por el algoritmo de acceso léxico, combinando costes del algoritmo en sí y costes adicionales estimados a partir de parámetros específicos. Con los resultados obtenidos, el uso del grafo no es una alternativa competitiva con respecto al árbol, de modo que en esta línea se propone en primer lugar la determinación de estrategias de ponderación más potentes, usando más fuentes de información en el proceso, así como el estudio de la posibilidad de hacer una estimación exacta de probabilidades para cada palabra. En este último caso habría que almacenar referencias adicionales en el proceso, aquellas que permitieran recuperar las historias perdidas en los puntos en que se toman decisiones.

En lo que respecta a estimación paramétrica de longitudes variables de listas de preselección y estimación no paramétrica basada en tablas de corte, queda para trabajos futuros el desarrollar más esta línea de investigación, al no haber sido posible abordarla en esta tesis con el rigor adecuado. De entrada habría que proceder a una evaluación rigurosa, complementada con el uso de distintas funciones y un número más amplio de parámetros. La metodología de trabajo propuesta para los estudios con redes neuronales sería perfectamente aplicados a este caso. En el caso concreto del cálculo de tablas de corte, proponemos como estrategia de trabajo el suavizado de los resultados usando *splines*, por ejemplo, para hacer frente al problema de granularidad y el de ajuste a los datos de entrenamiento.

En los procesos de estimación de longitudes de listas de preselección basadas en redes neuronales, la extensión inmediata al discriminador simple sería diseñar una estructura jerárquica de redes, de forma que entrenemos discriminadores aplicados sucesivamente y, posiblemente, apoyados en los resultados del discriminador anterior, de forma que aseguremos en todos los casos tareas que distingan únicamente entre dos alternativas y además estén razonablemente equilibradas en cuanto a número de ejemplos.

También se propone como línea futura el estudio de la modificación de los criterios de estimación de error en las redes neuronales para adecuarlos al problema que nos ocupa y mejorar la función objetivo en el entrenamiento de los pesos: la medida de error que nos interesaría usar debería tener relación con la tasa de inclusión y el esfuerzo medio obtenido.

A pesar de las consistentes mejoras obtenidas con los métodos basados en redes neuronales, no hemos podido demostrar la fiabilidad estadística de las mismas por falta de datos, con lo que uno de los objetivos a abordar en el futuro sería el análisis de dichos métodos con bases de datos mayores.

En las propuestas referidas a estimación de fiabilidad de hipótesis proponemos como línea de trabajo futuro la aplicación de las ideas vistas aquí en cuanto al uso de parámetros relacionados con el módulo de preselección en tareas de estimación de fiabilidad en sistemas completos de hipótesis-verificación, combinando aquellos con los obtenibles del proceso de búsqueda del módulo de análisis fino. Trabajos actualmente en curso en nuestro Grupo muestran la potencia de esta estrategia y están consiguiendo excelentes resultados en tareas de habla continua [SanSegundo01].

7.3 Sobre selección de unidades y diccionarios

En el apartado de selección de unidades, proponemos como tareas futuras la incorporación de unidades propuestas recientemente, como el semifonema (*demiphone*) [Mariño00], con la que se han conseguido excelentes resultados especialmente especialmente cuando los datos de entrenamiento son escasos. Igualmente se deberían abordar criterios de máxima verosimilitud para la selección, tanto del repertorio de unidades como de las alternativas de pronunciación, al estilo de las propuestas que se incluyen en [Holter98].

En la tesis se ha desarrollado algorítmica para el trabajo con grafos genéricos como soporte de múltiples pronunciaciones, en las que el proceso de corrección se implementa directamente sobre el grafo, lo que ayuda a compactar aún más la incorporación de alternativas. Sin embargo no se ha desarrollado esta idea, por lo que proponemos su uso como alternativa a la duplicación de entradas que se usa en la implementación actual. Esta estrategia, combinada con un entrenamiento de los costes de alineamiento del acceso léxico sobre la misma estructura de árbol permitirá, previsiblemente, conseguir mejores resultados que los obtenidos en la actualidad (esta idea va en la misma línea que el uso de costes contextuales propuesta en el apartado anterior).

En lo que respecta al uso de múltiples pronunciaciones y a pesar de los avances recientes en este tema [Strik99] y las propuestas de esta tesis, este área está aún en su infancia y el mayor problema con el que se enfrenta es la disponibilidad de bases de datos con un etiquetado específico mucho más fino que el disponen las actuales. Nuestra propuesta de continuación en este sentido sería el lanzamiento de una iniciativa de grabación extendida a todo el territorio nacional o, en su defecto, un etiquetado más preciso de las existentes.

La propuesta de combinación de métodos de generación de variantes de pronunciación basados en conocimiento y dirigidos por datos no ha sido validada por la ausencia de una base de datos adecuada. La consecución de la misma, en la línea de lo propuesto en el apartado anterior, ayudaría a realizar dicha evaluación

En cuanto a la selección de reglas del método basado en conocimiento, se podría plantear un mecanismo automático de generación de las mismas, como en [Hoste00], que se ajustaría perfectamente a las estructuras de búsqueda que usamos. Nuestra idea en este caso sería extraer y generalizar los comportamientos observados en las correcciones propuestas por el método dirigido por datos, seleccionando aquellos que con más frecuencia se aplican. Ello nos permitiría resolver uno de los grandes problemas de los métodos dirigidos exclusivamente por datos: la no aplicabilidad a diccionarios distintos a los de la base de datos de entrenamiento.

La aplicación de técnicas dirigidas por datos en el caso del sistema integrado no ha obtenido buenos resultados y queda pendiente una evaluación más detallada de los motivos, así como el análisis de la misma estrategia de corrección y filtrado usando los mismos modelos que en el sistema más potente, esto es, los semicontinuos dependientes del contexto.

La metodología de introducción de múltiples pronunciaciones que proponemos para análisis y validación futura como resultado de nuestro trabajo, sería el uso simultáneo de métodos dirigidos por datos y métodos basados en conocimiento para los procesos de generación de variantes, y los dirigidos por datos para los procesos de filtrado (validación). Es en esta sinergia de métodos donde creemos está la clave para el desarrollo de sistemas que integren múltiples pronunciaciones en el futuro. Esta propuesta se verá, de nuevo, dificultada por la carestía de bases de datos suficientemente amplias y con el suficiente detalle de etiquetado para permitir un análisis profundo de los efectos observados, y habrá que hacer un intenso esfuerzo en esa línea.

Una propuesta adicional de las medidas de diferencia de coste de acceso léxico es el diagnóstico y asesoramiento sobre la calidad y/o dificultad de diccionarios dados. En vocabularios pequeños es planteable un diseño manual, pero cuando se usan grandes diccionarios, es imprescincible contar con alguna metodología de diagnóstico más objetiva y que permita realizar el proceso automáticamente. A partir de un diccionario, sería posible identificar las palabras más problemáticas en cuanto a confusabilidad, seleccionando aquellas con la menor distancia. Dicha medida es más efectiva todavía si usamos costes de alineamiento entrenados para el tipo de modelado considerado. Esta estrategia de medida valdría igualmente para identificar elementos problemáticos de un diccionario, de cara a robustecer su modelado o, incluso, establecer modelos específicos para aquellos. Una aplicación inmediata de esta idea sería el diseño de vocabularios artificiales para la estimación de $\psi_i'(\lambda, V_i)$, lo que nos permitiría estudiar mejores aproximaciones que la pesimista que hemos utilizado en el desarrollo del Apartado 3.4.3.1 a partir de la página 62. Igualmente sería necesario estudiar más ampliamente las correlaciones estudiadas y analizar en profundidad el porqué de la falta de la misma con algunas de las medidas que, a priori, parecen adecuadas para evaluar la dificultad de un diccionario.

Por último, proponemos aplicar las ideas sobre complejidad de diccionarios y medidas de dificultad a la predicción de tasas de reconocimiento sobre diccionarios no vistos, en la línea de los trabajos descritos en [Roe94] y [Laface94].

A Parámetros de preselección

A.1 Introducción

En este apartado describiremos los parámetros que el módulo de acceso léxico genera como base para la toma de decisiones acerca de la longitud de lista de preselección a utilizar. La idea detrás de la selección que presentamos era tener el mayor número de parámetros posibles de entre los disponibles. Así, además de aquellos directamente relacionados con la palabra a reconocer (número de tramas, coste de reconocimiento acústico, etc.), se incluyen otros que pretenden reflejar la distribución estadística de los costes de acceso léxico para toda la lista de palabras. La motivación de incluir parámetros referentes a dicha distribución estadística vino dada precisamente por nuestro intento de disponer del mayor número de parámetros *razonables* posibles, aparte de los directamente obtenibles. Dicha decisión es también consistente con el uso por parte de algunos autores de medidas relacionadas con la diferencia en costes (o probabilidades) entre dos candidatos con el objeto de medir la fiabilidad de las hipótesis acústicas.

A.2 Descripción de los parámetros utilizados

El número total de parámetros generados es 35, aunque 2 de ellos no son utilizables en los algoritmos de estimación de longitud de lista, como se detalla más adelante. Todos ellos se describen en la Tabla A-1, pudiéndose distinguir cuatro grandes grupos:

- Parámetros directos: Directamente obtenibles de datos de la ocurrencia acústica a reconocer o del proceso de preselección: número de tramas, longitud de la cadena fonética, coste del algoritmo de búsqueda acústica, número de símbolos en el diccionario del primer candidato reconocido en el acceso léxico, coste del acceso léxico para dicho candidato.
- Parámetros derivados: A partir de los anteriores, aplicando normalizaciones de distinto tipo (dividiendo por el número de tramas, por la longitud de cadena fonética, etc.): coste acústico normalizado por la longitud de palabra o de cadena; coste del acceso léxico para el primer candidato normalizado por número de tramas, longitud de cadena o número de símbolos en el diccionario de dicho candidato; longitud de cadena normalizada por el número de tramas, etc.
- Parámetros estadísticos: Calculados sobre la distribución de los costes de acceso léxico, para distintas longitudes de la lista de preselección. Así, se calculan medias y desviaciones de dichos costes, normalizados o no según los criterios vistos más arriba, para longitudes iguales al 0'1%, 1%, 10%, 25% y 50% del tamaño del diccionario usado..

Tabla A-1: Parámetros disponibles para la estimación de longitudes variables de listas de preselección

Nº	Nombre del parámetro	Descripción
1	NumTramas	Número de tramas de la palabra a reconocer
2	NumSimbDic	Número de símbolos de la palabra a reconocer. *Evidentemente este parámetro no está disponible en estimación.*
3	LongLattice	Longitud de la cadena fonética o malla generada por el algoritmo de un paso
4	CostePSBU	Coste (log-probabilidad) estimado por el algoritmo de un paso

Tabla A-1: Parámetros disponibles para la estimación de longitudes variables de listas de preselección

Nº	Nombre del parámetro	Descripción
5	PosicOK	Posición en la que se reconoció la palabra. *Evidentemente este parámetro no está disponible en estimación, pero se incluye para facilitar cálculos posteriores*
6	NumSimb1erCand	Número de símbolos en el diccionario que tiene el candidato reconocido en primera posición (primer candidato)
7	CosteAL1erCand	Coste del acceso léxico para el primer candidato
8	CostePSBUNormNT	Coste (log-probabilidad) estimado por el algoritmo de un paso, normalizado por el número de tramas de la palabra
9	CostePSBUNormLL	Coste (log-probabilidad) estimado por el algoritmo de un paso, normalizado por la longitud de la cadena fonética o malla
10	CosteAL1erCandNormNT	Coste de acceso léxico del primer candidato normalizado por el número de tramas
11	CosteAL1erCandNormLL	Coste de acceso léxico del primer candidato normalizado por la longitud de la cadena fonética o malla
12	CosteAL1erCandNormNS1	Coste de acceso léxico del primer candidato normalizado por el número de símbolos del diccionario del mismo
13	LongLatNumTramas	Cociente entre la longitud de la cadena fonética y el número de tramas
14	CostePSBUNormNS1	Coste (log-probabilidad) estimado por el algoritmo de un paso, normalizado por el número de símbolos en el diccionario del primer candidato
15	NumSimb1erCandLongLat	Cociente entre el número de símbolos en el diccionario del primer candidato y la longitud de la cadena fonética
16	Media01CosteAL	Media de los costes de acceso léxico para un número de candidatos igual al 0.1% del tamaño del diccionario
17	Desv01CosteAL	Desviación de los costes de acceso léxico para un número de candidatos igual al 0.1% del tamaño del diccionario
18	Media01CosteALNormLL	Media de los costes de acceso léxico normalizados por longitud de cadena fonética para un número de candidatos igual al 0.1% del tamaño del diccionario
19	Desv01CosteALNormLL	Desviación de los costes de acceso léxico normalizados por longitud de cadena fonética para un número de candidatos igual al 0.1% del tamaño del diccionario
20	Media1CosteAL	Media de los costes de acceso léxico para un número de candidatos igual al 1% del tamaño del diccionario
21	Desv1CosteAL	Desviación de los costes de acceso léxico para un número de candidatos igual al 1% del tamaño del diccionario
22	Media1CosteALNormLL	Media de los costes de acceso léxico normalizados por longitud de cadena fonética para un número de candidatos igual al 1% del tamaño del diccionario
23	Desv1CosteALNormLL	Desviación de los costes de acceso léxico normalizados por longitud de cadena fonética para un número de candidatos igual al 1% del tamaño del diccionario
24	Media10CosteAL	Media de los costes de acceso léxico para un número de candidatos igual al 10% del tamaño del diccionario

Tabla A-1: Parámetros disponibles para la estimación de longitudes variables de listas de preselección

Nº	Nombre del parámetro	Descripción
25	Desv10CosteAL	Desviación de los costes de acceso léxico para un número de candidatos igual al 25% del tamaño del diccionario
26	Media10CosteALNormLL	Media de los costes de acceso léxico normalizados por longitud de cadena fonética para un número de candidatos igual al 10% del tamaño del diccionario
27	Desv10CosteALNormLL	Desviación de los costes de acceso léxico normalizados por longitud de cadena fonética para un número de candidatos igual al 10% del tamaño del diccionario
28	Media25CosteAL	Media de los costes de acceso léxico para un número de candidatos igual al 25% del tamaño del diccionario
29	Desv25CosteAL	Desviación de los costes de acceso léxico para un número de candidatos igual al 25% del tamaño del diccionario
30	Media25CosteALNormLL	Media de los costes de acceso léxico normalizados por longitud de cadena fonética para un número de candidatos igual al 251% del tamaño del diccionario
31	Desv25CosteALNormLL	Desviación de los costes de acceso léxico normalizados por longitud de cadena fonética para un número de candidatos igual al 25% del tamaño del diccionario
32	Media50CosteAL	Media de los costes de acceso léxico para un número de candidatos igual al 50% del tamaño del diccionario
33	Desv50CosteAL	Desviación de los costes de acceso léxico para un número de candidatos igual al 50% del tamaño del diccionario
34	Media50CosteALNormLL	Media de los costes de acceso léxico normalizados por longitud de cadena fonética para un número de candidatos igual al 50% del tamaño del diccionario
35	Desv50CosteALNormLL	Desviación de los costes de acceso léxico normalizados por longitud de cadena fonética para un número de candidatos igual al 50% del tamaño del diccionario

B Bases de datos y tareas

B.1 Introducción

En este apéndice se describe el contenido de cada una de las bases de datos utilizadas, así como los nemotécnicos que utilizamos en este documento al referirnos a ellas, o a subconjuntos de ellas.

B.2 VESTEL (TIDAISL)

B.2.1 Descripción general

VESTEL es una base de datos telefónica capturada sobre la red telefónica pública y que está compuesta de dígitos, números, comandos, nombres propios, etc. y diseñada para soportar investigación y desarrollo en sistemas de reconocimiento automático de habla, con independencia del locutor y basada en unidades inferiores a la palabra.

B.2.2 Contenido

La parte de VESTEL que hemos utilizado define tres subconjuntos fundamentales de trabajo:

- PRNOK: Conjunto destinado al entrenamiento genérico de los sistemas que operan sobre VESTEL. Está compuesto por 5820 ficheros, conteniendo 1175 palabras (grafemas) distintos, y 3011 locutores diferentes.
- PERFDV: Conjunto destinado al reconocimiento e inicialmente diseñado para realizar pruebas "dependientes del vocabulario", en el sentido de que comparte con PRNOK los grafemas. Está compuesto por 2536 ficheros, conteniendo 419 palabras distintas, pronunciadas por 2255 locutores diferentes.
- PEIV1000: Conjunto destinado al reconocimiento e inicialmente diseñado para realizar pruebas "independientes del vocabulario", en el sentido de que no comparte con PRNOK prácticamente ningún grafema (como se indica en el Apartado B.2.5, realmente hay 7 grafemas comunes). Está compuesto por 1434 ficheros, conteniendo 781 palabras distintas pronunciadas por 1351 locutores diferentes

En los experimentos descritos en esta tesis haremos referencia a la tarea VESTEL como aquella que analiza el resultado sobre PERFDV y/o PEIV1000 , usando obviamente los modelos entrenados con PRNOK5TR. Dicha tarea se usará en experimentos preliminares, en aquellos en los se quieran sacar conclusiones sobre dependencia o independencia de vocabulario y en aquellos en los que se requiera su uso por razones concretas (prefiriéndola a la descrita a continuación).

Además de estos subconjuntos definidos, se realizó una partición adicional utilizando la técnica de *leave-one-out* (VESTEL-L) para incrementar la fiabilidad de los resultados de reconocimiento obtenidos y aportar una visión distinta de la tarea. Para ello:

- Se generó una lista compuesta por todos los subconjuntos definidos anteriormente (PRNOK, PERFDV y PEIV1000), haciendo un total de 9756 ficheros
- Se generó un total de 10 particiones de esa lista unificada, asignando por convención el siguiente criterio de nombrado para cada una de ellas:
  - `loo.trn-#`: Lista de entrenamiento correspondiente a la partición número # (# = 0..9) compuesta por alrededor de 8874 ficheros.

- `loo.tst-#`: Lista de reconocimiento correspondiente a la partición número # (# = 0..9), compuesta de alrededor de 882 ficheros.

En los experimentos descritos en esta tesis haremos referencia a la tarea VESTEL-L como aquella que analiza el resultado medio para las 10 particiones de TIDAISL (`loo.tst-[0..9]`), usando obviamente los modelos entrenados con la partición de entrenamiento asociada correspondiente (`loo.trn-[0..9]`). En esta tarea no se hacen consideraciones sobre la dependencia o independencia de vocabulario sino que, como se ha dicho, se trata fundamentalmente de evaluar la complejidad de la misma y de obtener resultados con mayor fiabilidad estadística.

B.2.3 Diccionarios

En las tareas de reconocimiento abordadas se han diseñado 2 grupos de diccionarios distintos, a saber:

- Diccionarios para las listas básicas (PRNOK, PERFDV y PEIV1000)
  - DV1175: Compuesto por 1175 palabras, diseñado para soportar tareas sobre PRNOK y PERFDV
  - DV5000: Compuesto por las 1175 palabras de DV1175 completadas hasta el total por palabras provenientes de ONOMASTICA y con una distribución de longitudes medias similar al diccionario base[1]
  - DV10000: Compuesto por las 5000 palabras de DV5000 completadas hasta el total por palabras provenientes de ONOMASTICA
  - IV1996: Compuesto por 1996 palabras, diseñado para soportar tareas sobre PEIV1000
  - IV5000: Compuesto por las 1996 palabras de IV1996 completadas hasta el total por palabras provenientes de ONOMASTICA
  - IV10000: Compuesto por las 5000 palabras de IV5000 completadas hasta el total por palabras provenientes de ONOMASTICA
- Diccionarios para las listas procedentes del particionado descrito (VESTEL-L):
  - VESTEL-L1952: Compuesto por las 1952 palabras distintas que aparecen en las tres listas principales
  - VESTEL-L5000-50-50: Compuesto por VESTEL-L1952 al que se añadieron palabras procedentes de DV5000 e IV5000 en la misma proporción (de las 3048 necesarias, 1524 proceden de DV5000 y 1524 de IV5000).
  - VESTEL-L5000-85-15: Compuesto por VESTEL-L1952 al que se añadieron palabras procedentes de DV5000 e IV5000 en la misma proporción que la composición que la base de datos original que se particionó (de las 9720 palabras, aproximadamente un 85% corresponden a un vocabulario con características de longitud similares a DV1175 y un 15% a un vocabulario con características de longitud similares a IV1996) (de las 3048 necesarias, 2621 proceden de DV5000 y 427 de IV5000).
  - VESTEL-L10000-50-50: Compuesto por VESTEL-L5000-50-50 y palabras procedentes de DV10000 e IV10000 en la misma proporción que se vio para VESTEL-L5000-50-50.
  - VESTEL-L10000-85-15: Compuesto por VESTEL-L5000-85-15 y palabras procedentes de DV10000 e IV10000 en la misma proporción que se vio para VESTEL-L5000-85-15.

1. Para ampliar detalles sobre el proceso de ampliación de diccionarios, remitimos al lector al Anexo C

B.2.4 Tareas

Las tareas de reconocimiento abordadas son las que corresponden a la combinación de cualquiera de las bases de datos definidas en el Apartado B.2.2, esto es VESTEL (PRNOK, PERDV y PEIV1000) y VESTEL-L, con los diccionarios susceptibles de utilización descritos en el Apartado B.2.3.

B.2.5 Información cuantitativa adicional: Estadísticas comparativas y distribución de ocurrencias

Se incluyen a continuación algunas cifras comparativas de las intersecciones entre los segmentos de las bases de datos descritos en este apartado, en lo que se refiere a locutores y palabras (grafemas) ya que, evidentemente no hay ningún fichero compartido entre ninguna de las tres partes ni tomadas de dos en dos.

En la Tabla B-1 se indican los grafemas compartidos por las listas, tomadas dos a dos junto con el número total de ficheros y de grafemas distintos en cada una de ellas.

Tabla B-1: Grafemas comunes en los subconjuntos de datos de VESTEL

	Número total de ficheros	**Número total de grafemas distintos en la lista**	prnok	perfdv	peiv1000
prnok	5820	1175		419	7
perfdv	2536	419	419		0
peiv1000	1434	781	7	0	

En la Tabla B-2 se indican los locutores compartidos por las listas, tomadas dos a dos junto con el número total de ficheros y de grafemas distintos en cada una de ellas

Tabla B-2: Locutores comunes en los subconjuntos de datos de VESTEL

	Número total de ficheros	**Número total de locutores distintos en la lista**	prnok	perfdv	peiv1000
prnok	5820	3011		472	335
perfdv	2536	2255	472		357
peiv1000	1434	1351	335	357	

Con las cifras vistas más arriba, disponemos aproximadamente

Para hacernos una idea de la complejidad de las tareas de VESTEL con las que nos enfrentamos, hemos calculado la distribución estadística de las repeticiones por palabra disponibles en los casi 10000 ficheros de los que disponemos.

En la Figura B-1 se muestra el porcentaje de palabras de la base de datos en función del número de repeticiones disponibles (línea continua), así como el histograma de porcentaje acumulado (línea discontinua). Como puede verse claramente, alrededor del 60% de las palabras disponibles tienen

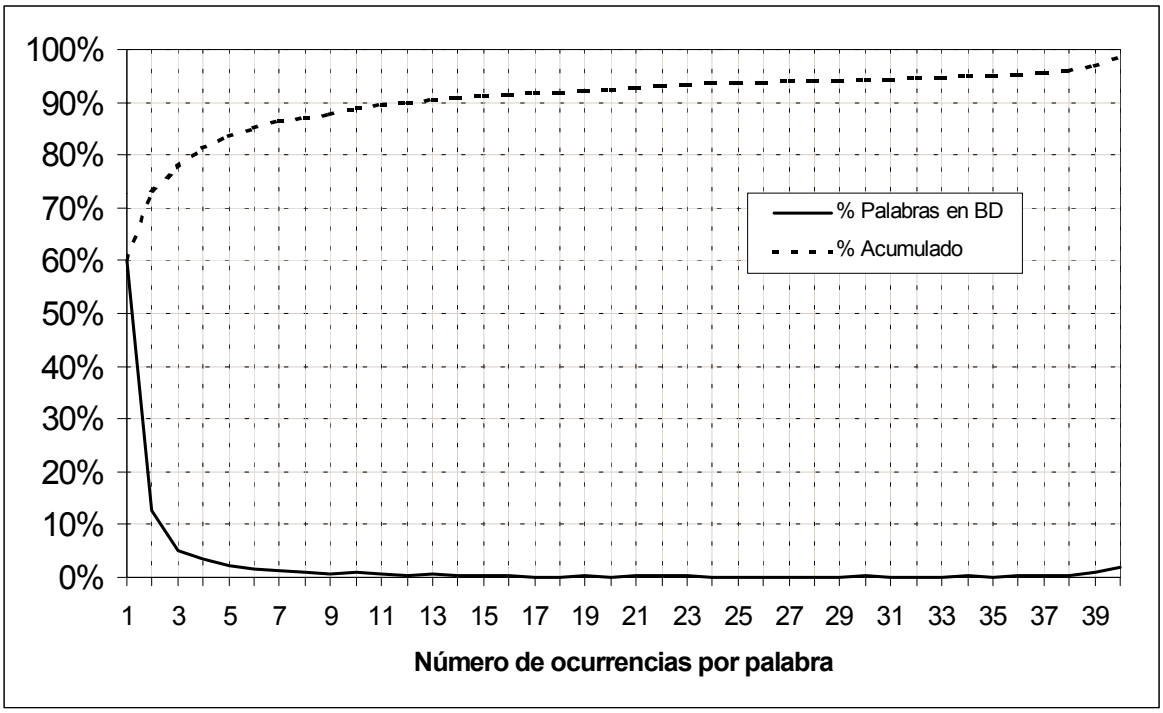

Figura B-1: Distribución de ocurrencias por palabra en la base de datos VESTEL (PRNOK+PERFDV+PEIV1000)

una sola repetición, al tiempo que si consideramos las palabras con menos de 10 repeticiones, cubrimos casi un 90% del total. Las distribuciones individuales para PRNOK, PERFDV y PEIV1000 son muy similares a la vista.

En el Anexo C se incluye información cuantitativa específica acerca de la composición y características de los diccionarios usados.

B.3 POLYGLOT

B.3.1 Descripción general

POLYGLOT es una base de datos de habla limpia y que está compuesta por palabras genéricas procedentes de un corpus de textos periodísticos y diseñada para soportar investigación y desarrollo en sistemas de reconocimiento automático de habla, con dependencia del locutor y basada en unidades inferiores a la palabra.

B.3.2 Contenido

POLYGLOT está compuesta por 30000 palabras y 30 locutores, habiéndose grabado 1000 palabras por locutor, divididas en dos conjuntos de datos:

- set-a: Compuesto por 500 palabras por locutor, utilizada como base de datos de entrenamiento
- set-c: Compuesto por 500 palabras por locutor, utilizado como base de datos de reconocimiento.

POLYGLOT fue grabada en dos etapas, generando dos subconjuntos de datos con características ligeramente diferentes:

- CONJUNTO POLYGLOT-1: Grabaciones correspondientes a 10 locutores españoles nativos, 5 hombres y 5 mujeres. Para cada locutor se grabaron los dos conjuntos de datos descritos anteriormente, lo que hace un total de 10000 palabras. Las grabaciones se realizaron en una habitación silenciosa usando un equipo REVOX B77 MkII (cinta magnética) y un micrófono RCF MD 2700. La digitalización se llevó a cabo con el sistema de conversión de datos de audio Digital Sound Corporation DSC-200, usando una frecuencia de muestreo de 16KHz, un filtro paso bajo de 6.4 KHz de frecuencia de corte superior y un tamaño de palabra de 16 bits (formato Intel). En la Tabla B-2 se

Tabla 7-1: Locutores de POLYGLOT-1

Código	Sexo
lui	Hombre
cel	Mujer
emi	Mujer
gut	Hombre
est	Mujer
lea	Hombre
mor	Hombre
qui	Hombre
san	Hombre
sus	Mujer

indican los códigos de los locutores disponibles, así como su sexo.

Las 500 palabras correspondientes al set-a están segmentadas y etiquetadas fonéticamente, y las marcas manuales de segmentación se utilizan en el proceso de entrenamiento.

Las 500 palabras correspondientes al set-c se segmentaron automáticamente utilizando un detector de principio y fin basado en umbrales de energía y heurísticos temporales.

- CONJUNTO POLYGLOT-2: Grabaciones correspondientes a 20 locutores españoles nativos, 10 hombres y 10 mujeres. Para cada locutor se grabaron los dos conjuntos descritos anteriormente, lo que hace un total de 20000 palabras. Las grabaciones se realizaron en una sala sorda digitalizando directamente las mismas, usando una tarjeta de conversión OROS-22, usando una frecuencia de muestreo de 16KHz, un filtro paso bajo de 6.4 KHz de frecuencia de corte superior y un tamaño de palabra de 16 bits (formato

Intel). El micrófono usado fue de nuevo el RCF

Tabla 7-2: Locutores de POLYGLOT-2

Código	Sexo
sfa	Mujer
jvp	Hombre
mga	Hombre
jfl	Hombre
aga	Mujer
ogg	Hombre
npn	Mujer
jpm	Hombre
rch	Hombre
efc	Hombre
aa	Hombre
mgb	Hombre
mag	Mujer
gcg	Hombre
ega	Hombre
pag	Mujer
mab	Hombre
aag	Mujer
jih	Hombre
fgg	Hombre

MD 2700. En la Tabla B-2 se indican los códigos de los locutores disponibles, así como su sexo.

Las 1000 palabras por locutor correspondientes a los dos conjuntos de datos se segmentaron automáticamente utilizando un detector de principio y fin basado en umbrales de energía y heurísticos temporales.

B.3.3 Diccionarios

En las pruebas realizadas se utilizó un diccionario de 2000 palabras, seleccionado en el proyecto POLYGLOT y compuesto por las 1000 palabras distintas de las que se disponen grabaciones a las que se añadieron otras mil, usando un criterio de frecuencia de uso en el corpus de textos de partida.

B.3.4 Tareas

La tarea que planteamos en esta tesis sobre POLYGLOT es dependiente del locutor, trabajando con los 30 locutores por separado y ofreciendo los resultados medios para todos ellos.

Es importante destacar que las diferencias cualitativas entre POLYGLOT-1 y POLYGLOT-2, en lo que se refiere a la ausencia de una segmentación manual de los subconjuntos de entrenamiento dan lugar a tasas de reconocimiento menores en el segundo caso, a pesar de lo cual se ha preferido

mantener el análisis conjunto con todos los locutores, con el objetivo de aumentar la fiabilidad estadística de los resultados.

Por último, indicar que no se planteó una tarea multilocutor al considerarla poco importante de cara a los objetivos de la tesis.

C Consideraciones sobre la ampliación de diccionarios en VESTEL

C.1 Introducción

En este apéndice trataremos con un cierto detalle el tema de la ampliación del vocabulario para las tareas con 5000 y 10000 palabras, y los conceptos de dependencia e independencia del vocabulario.

C.2 Ampliación de diccionarios

En TIDAISL se dispone de diccionarios originales, extraídos de la aplicación, con 1175 y 1996 palabras, para tareas dependientes e independientes del vocabulario, respectivamente.

Tras los primeros resultados del sistema de preselección, se planteó que posiblemente no tuviera sentido utilizar el mismo en tareas de 1000 o 2000 palabras, de modo que se sugirió la realización de experimentos con vocabularios mucho mayores.

El problema fundamental de esta propuesta es la búsqueda y selección de las palabras que van a completar los diccionarios existentes.

En nuestro caso, optamos por completarlos utilizando palabras de la base de datos generada en el proyecto europeo ONOMASTICA, en el que el Grupo había estado involucrado como uno de los socios. En ONOMASTICA se disponía de diccionarios de nombres propios, topónimos, etc, con hasta 60000 entradas, justamente del mismo dominio semántico que las palabras de la base de datos VESTEL, con la que hemos estado trabajando.

La selección se hizo intentando mantener la "filosofía" de los diccionarios existentes, en el sentido de que la lista dependiente del vocabulario estaba compuesta de nombres simples, y la independiente, de nombres compuestos.

Así, creamos diccionarios adicionales de 5000 y 10000 palabras, completando en cada caso los diccionarios originales de 1175 y 1996 palabras. Esto quiere decir que nuestro inventario final de diccionarios es como sigue:

- 1175 palabras originales. Tarea dependiente del vocabulario
- 1196 palabras originales. Tarea independiente del vocabulario
- 1175 palabras originales + 3825 palabras de ONOMASTICA (nombres simples) = 5000 palabras, para experimentos con dependencia del vocabulario[1]
- 1175 palabras originales + 8825 palabras de ONOMASTICA (nombres simples) = 10000 palabras, para experimentos con dependencia del vocabulario
- 1996 palabras originales + 3004 palabras de ONOMASTICA (nombres compuestos) = 5000 palabras, para experimentos con independencia del vocabulario
- 1996 palabras originales + 8004 palabras de ONOMASTICA (nombres compuestos) = 10000 palabras, para experimentos con dependencia del vocabulario

1. No vamos a entrar ahora en consideraciones sobre esto y lo trataremos un poco más adelante

C.3 Consideraciones sobre dependencia e independencia del vocabulario

En este punto hay que reflexionar sobre la calificación de "dependencia" o "independencia" del vocabulario.

Está claro que en las listas originales, los conceptos de dependencia e independencia son aplicables y están claros. Sin embargo, en el momento en el que "artificialmente" añadimos palabras de otra fuente, la aplicabilidad deja de tener sentido, por lo menos si hablamos estrictamente.

Por ejemplo, es evidente que en caso de diccionarios de 5000 y 10000 palabras generados al completar las 1175 de dependencia del vocabulario, no podemos seguir hablando de "dependencia", ya que hemos introducido palabras no vistas en el set de entrenamiento.

Lo mismo aplicaría al caso de los diccionarios resultantes de completar los de 5000 y 10000 a partir de las 1996 palabras, aunque en este caso estamos prácticamente seguros de que no hay ninguna palabra presente en el set de entrenamiento.

Un problema adicional surge en la extracción de las palabras adicionales. Hemos verificado que el sistema funciona mejor en términos relativos (mejor tasa de reconocimiento para la misma reducción de vocabulario) en los experimentos con independencia que con dependencia, a pesar de que la tarea es, en principio, más compleja.

Dicho resultado lo hemos atribuido a la menor similaridad acústica de las palabras presentes en los diccionarios y listas independientes del vocabulario, y uno de los factores que pueden influir en esto es la longitud media de las mismas (medidas en número de alófonos): a mayor longitud, más información disponible, con lo que el reconocimiento puede ser más "fácil".

A continuación incluimos los resultados de un estudio que hicimos al respecto.

C.4 Estudio sobre longitudes medias de listas y diccionarios

Las cifras que vamos a dar se refieren a:

- Longitud media de la lista: Longitud media de la base de datos completa.
- Longitud media del diccionario: Longitud media del diccionario de que se trate.

C.4.1 Prnok5tr

Longitud media lista	6.415
Longitud media diccionario 1175	6.060
Longitud media diccionario 5000DV	6.839
Longitud media diccionario 10000DV	6.942

C.4.2 Perfdv

Longitud media lista	5.867
Longitud media diccionario 1175	6.060
Longitud media diccionario 5000DV	6.839
Longitud media diccionario 10000DV	6.942

C.4.3 Peiv1000

Longitud media lista	8.174
Longitud media diccionario 1996	7.521
Longitud media diccionario 5000IV	8.990
Longitud media diccionario 10000IV	9.488

C.4.4 Variaciones relativas

En la Tabla C-1 se presentan los resultados de comparar los valores vistos más arriba entre sí, dando un porcentaje de incremento o decremento en la longitud de cada uno. Por ejemplo, la media de longitud de las palabras de la lista `prnok5tr` (en horizontal, primera fila: 6.415), es un 9.34% mayor que la de `perfdv` (en vertical, segunda columna: 5.867), y así sucesivamente. Resumiendo: un valor positivo en una casilla indica que la longitud media de la lista/diccionario indicada en la fila es superior en ese tanto por ciento a la entrada en la columna correspondiente. Las cabeceras de filas o columnas sombreadas corresponden a listas de ficheros, y el resto a diccionarios (en cualquier caso, los nombres son auto-explicativos).

No vamos a entrar en un análisis detallado de los datos expuestos, que corroboran la idea de longitudes mayores en listas y diccionarios independientes del vocabulario. Simplemente plantearemos que puede ser necesario un estudio del efecto real que puede tener la longitud de las palabras de reconocimiento en el rendimiento del sistema.

Tabla C-1: Comparación de longitudes medias entre listas y diccionarios de la distribución (% de diferencia)

	prnok5	perfdv	peiv10	1175	1996	5000d	10000d	5000i	10000i
Media prnok5tr 6.415		9.34	-21.52	5.85	-14.71	-6.19	-7.59	-28.64	-32.39
Media perfd 5.867	-8.54		-28.22	-3.18	-21.99	-14.21	-15.49	-34.74	-38.16
Media peiv1 8.174	27.4	39.32		34.88	8.68	19.52	17.74	-9.08	-13.84
Media dic1175 6.060	-5.5	3.28	-25.86		-19.43	-11.39	-12.71	-32.59	-36.13
Media dic1996 7.521	17.24	28.19	-7.99	24.11		9.97	8.34	-16.24	-20.73
Media dic5000d 6.839	6.61	16.56	-16.33	12.85	-9.07		-1.48	-23.93	-27.92
Media dic10000d 6.94	8.22	18.32	-15.07	14.55	-7.69	1.51		-22.78	-26.83
Media dic5000i 6.990	40.14	53.23	9.98	48.55	19.53	31.45	29.50		-5.25
Media dic10000i 9.49	47.90	61.72	16.08	56.57	26.15	38.73	36.68	5.54	

D Alfabetos utilizados

D.1 Introducción

En este apéndice se describe el contenido de los alfabetos utilizados en todos los sistemas de reconocimiento utilizados en este trabajo de tesis.

En los siguientes apartados se incluye información de cada alfabeto seleccionado manual o automáticamente, indicando su nombre, los criterios utilizados en dicha decisión, así como estadísticas de su presencia en las bases de datos utilizadas.

En todos los alfabetos utilizados se amplió el repertorio de unidades en la experimentación, añadiendo dos modelos adicionales de silencio (inicial y final).

D.2 Alfabetos manuales

D.2.1 Alfabeto: `alf51`

Es el alfabeto más completo de los usados estando compuesto por 51 unidades y es el usado en los sistemas de conversión de texto a voz de nuestro Grupo. Ha servido de base para todos los sistemas de reconocimiento desarrollados, aunque a lo largo del tiempo ha sufrido ciertas modificaciones orientadas a hacerlo más adecuado a las tareas de reconocimiento.

La motivación de su uso inicial es la disponibilidad de un conversor grafema-fonema automático y muy sofisticado. Las adaptaciones posteriores son posibles gracias a programas de conversión de fácil manejo.

D.2.1.1 Contenido

Es el mostrado en la Tabla D-1.

Tabla D-1: Contenido del alfabeto `alf51`. Total: 51 unidades

Código	Símbolo	Descripción	Ejemplo
0	a	Vocales	cas**a**
1	e		esp**e**ra
2	i		hac**i**a
3	o		c**o**lega
4	u		**u**sted
5	'a	Vocales acentuadas	c**a**sa
6	'e		ac**e**ra
7	'i		m**i**to
8	'o		c**o**sa
9	'u		c**u**po

Tabla D-1: Contenido del alfabeto `alf51`. Total: 51 unidades

Código	Símbolo	Descripción	Ejemplo
10	a~	Vocales nasalizadas	**a**ñade
11	e~		**e**nseña
12	i~		**i**nfante
13	o~		**o**ndulado
14	u~		**u**ntado
15	'a~	Vocales acentuadas y nasalizadas	c**a**ma
16	'e~		c**e**na
17	'i~		c**i**ne
18	'o~		c**o**no
19	'u~		c**u**na
20	w	Semiconsonantes	ac**u**erdo
21	j		aci**e**rto
22	U	Semivocales	arau**j**o
23	I		arna**i**z
24	b	Oclusivas	**b**odega
25	d		**d**ado
26	g		**g**ato
27	p		**p**etaca
28	t		pe**t**aca
29	k		peta**c**a
30	f	Fricativas	**f**igura
31	X		e**j**emplo
32	T		a**c**ierto
33	s		**s**onido
34	B		a**b**ogado
35	D		aboga**d**o
36	G		abo**g**ado
37	J		ama**y**a
38	n	Nasales	a**n**a
39	N		be**n**goa
40	N~		a**ñ**o
41	m		a**m**o
42	r	Vibrantes	ca**r**o
43	R		ca**rr**o
44	R*		césa**r**
45	R/		c**r**espo
46	R_		is**r**ael
47	l	Laterales	ho**l**a
48	L		o**ll**a

Tabla D-1: Contenido del alfabeto `alf51`. Total: 51 unidades

Código	Símbolo	Descripción	Ejemplo
49	T/	Africadas	o**ch**o
50	J/		**y**untas

D.2.1.2 Estadísticas de ocurrencias

En la Figura D-1 y la Figura D-2 se incluye la distribución estadística de las ocurrencias de cada una de las unidades del alfabeto `alf51` en las bases de datos disponibles en VESTEL y POLYGLOT, respectivamente. Como puede observarse, dicha distribución es razonablemente homogénea en los conjuntos de entrenamiento y reconocimiento..

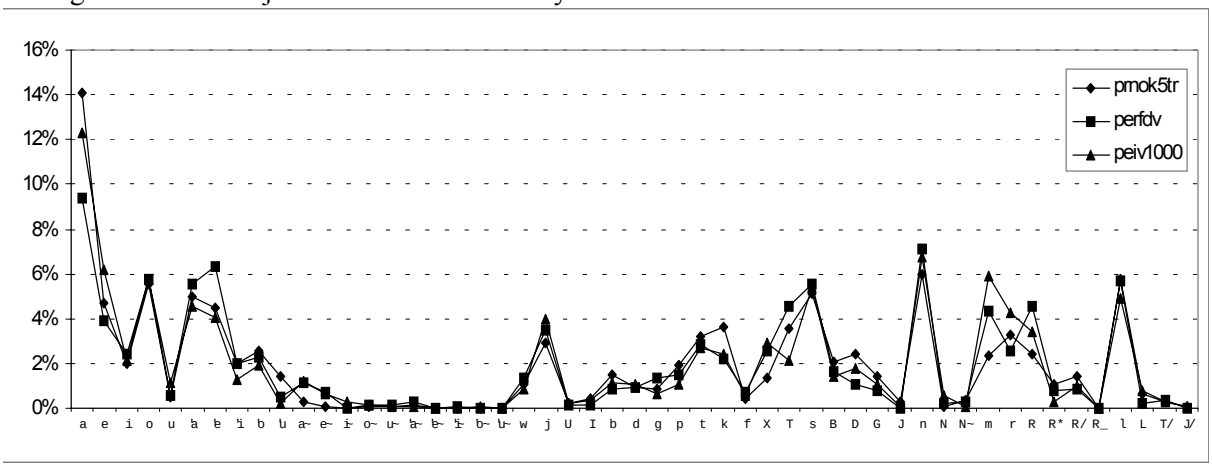

Figura D-1: Distribución de las ocurrencias de las unidades de `alf51` en las bases de datos de VESTEL

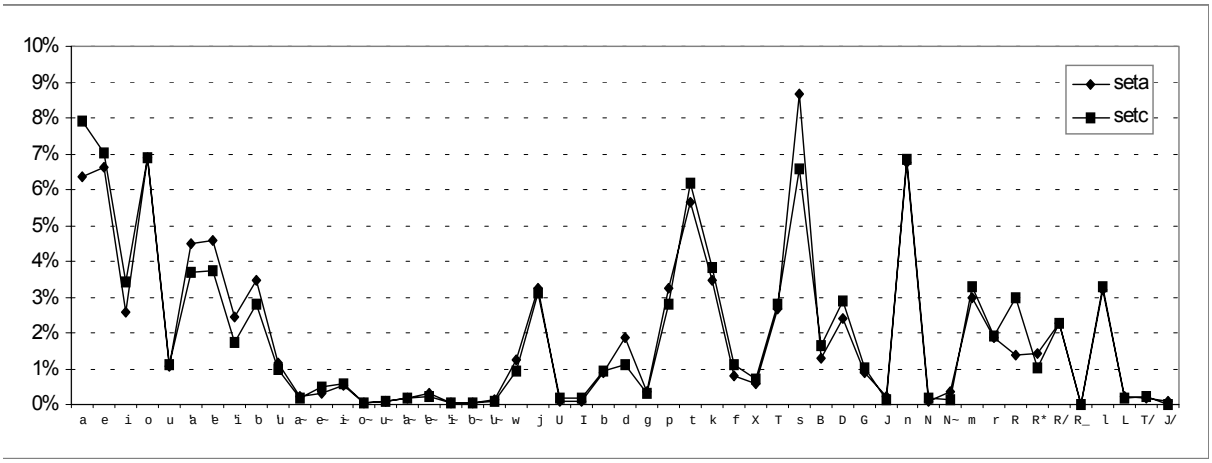

Figura D-2: Distribución de las ocurrencias de las unidades de `alf51` en los subconjuntos de POLYGLOT

Igualmente puede observarse una cierta deficiencia de ocurrencias para algunas unidades, lo que puede causarnos problemas al dar lugar a un entrenamiento poco robusto. Así, en lo que respecta al número global de ocurrencias en las listas de entrenamiento, incluimos en la Tabla D-2 los valores correspondientes a las unidades más escasas[1]. Como puede observarse, en algunos casos nos encontramos con valores muy preocupantes, al no disponer de más que unas pocas repeticiones (se han marcado en negrita las inferiores a 5 ocurrencias).

Tabla D-2: Número de unidades de `alf51` disponibles en las listas de entrenamiento con posibles problemas de número de repeticiones (posible entrenamiento deficiente)

ALF51	Símbolo	prnok5tr	set-a
10	a~		6
11	e~	36	
12	i~	9	
13	o~	21	**1**
14	u~	17	**3**
15	'a~	66	5
16	'e~	12	
17	'i~	12	**1**
18	'o~	**3**	**1**
19	'u~	**0**	**4**
22	U		**3**
23	I		**3**
39	N	33	**3**
46	R_	**1**	**0**
49	T/		5
50	J/	14	**2**

Este problema de falta de material de entrenamiento a priori es especialmente grave en el caso de las bases de datos de POLYGLOT. Así, el trabajo posterior se centró en conseguir alfabetos lo más completos posibles manteniendo un número mínimo de ocurrencias para todas las unidades (`alf45`, como se verá) y otros en los que el criterio fundamental era la simplicidad y la robustez de entrenamiento (`alf23` y `alf33`, como se verá).

D.2.2 Alfabeto: `alf45`

Procede de `alf51`, está compuesto por 45 unidades y es el resultado de agrupar algunas de las unidades menos entrenadas de las tareas sobre VESTEL, tratando de mantener la mayor resolución acústico-fonética posible.

1. El criterio de definición de *escasez* ha sido el considerar aquellas unidades cuyo número de ocurrencias es inferior al 10% del porcentaje que correspondería a una distribución uniforme (mismo número de ocurrencias para todas las posibles). Por ejemplo, en PRNOK disponemos de 37325 unidades pertenecientes a `alf51` (51 unidades), con lo que marcarán como insuficientes (a priori) aquellas que no superen el 10% de 37325/51, es decir, 78 repeticiones.

D.2.2.1 Contenido

Es el mostrado en la Tabla D-3.

Tabla D-3: Contenido del alfabeto `alf45`. Total: 45 unidades

Código	Símbolo	Descripción	Ejemplo
0	a	Vocales	cas**a**
1	e		esp**e**ra
2	i		hac**i**a
3	o		c**o**lega
4	u		**u**sted
5	'a	Vocales acentuadas	c**a**sa
6	'e		ac**e**ra
7	'i		m**i**to
8	'o		c**o**sa
9	'u		c**u**po
10	a~	Vocales nasalizadas, acentuadas o no	**a**ñade c**a**ma
11	e~		**e**nseña cena
12	i~		**i**nfante cine
13	o~		**o**ndulado cono
14	u~		**u**ntado cuna
15	w	Semiconsonantes	ac**u**erdo
16	j		ac**i**erto
17	U	Semivocales	arau**j**o
18	I		arna**i**z
19	b	Oclusivas	**b**odega
20	d		**d**ado
21	g		**g**ato
22	p		**p**etaca
23	t		pe**t**aca
24	k		peta**c**a

Tabla D-3: Contenido del alfabeto `alf45`. Total: 45 unidades

Código	Símbolo	Descripción	Ejemplo
25	f	Fricativas	**f**igura
26	X		e**j**emplo
27	T		a**c**ierto
28	s		**s**onido
29	B		a**b**ogado
30	D		aboga**d**o
31	G		abo**g**ado
32	J		ama**y**a
33	n	Nasales	a**n**a
34	N		be**n**goa
35	N~		a**ñ**o
36	m		a**m**o
37	r	Vibrantes	ca**r**o
38	R		ca**rr**o is**r**ael
39	R*		césa**r**
40	R/		c**r**espo
41	l	Laterales	ho**l**a
42	L		o**ll**a
43	T/	Africadas	o**ch**o
44	J/		**y**untas

Como puede observarse, al compararlo con la Tabla D-1, se han unificado las vocales nasalizadas, ya estén acentuadas o no, y se ha eliminado la R_ (R tras s), unificándola con la R doble convencional.

D.2.2.2 Estadísticas de ocurrencias

En la Figura D-3 y la Figura D-4 se incluye la distribución estadística de las ocurrencias de cada una de las unidades del alfabeto `alf45` en las bases de datos disponibles en VESTEL y POLYGLOT, respectivamente. Como puede observarse, dicha distribución es muy parecida a la de `alf51`.

Igualmente sigue observándose una cierta deficiencia para algunas unidades en las listas de POLYGLOT.

Así, en lo que respecta al número global de ocurrencias en las listas de entrenamiento, incluimos en la Tabla D-4 los valores correspondientes a las unidades más escasas[1]. Como puede observarse, las unidades más problemáticas en VESTEL cuentan ahora con un número de ocurrencias

1. El criterio de definición de escasez ha sido el considerar aquellas unidades cuyo número de ocurrencias es inferior al 10% del porcentaje que correspondería a una distribución uniforme (mismo número de ocurrencias para todas las posibles). Por ejemplo, en PRNOK disponemos de 37325 unidades pertenecientes a `alf51` (51 unidades), con lo que marcarán como insuficientes (a priori) aquellas que no superen las 37325/51/10 = 78 unidades.

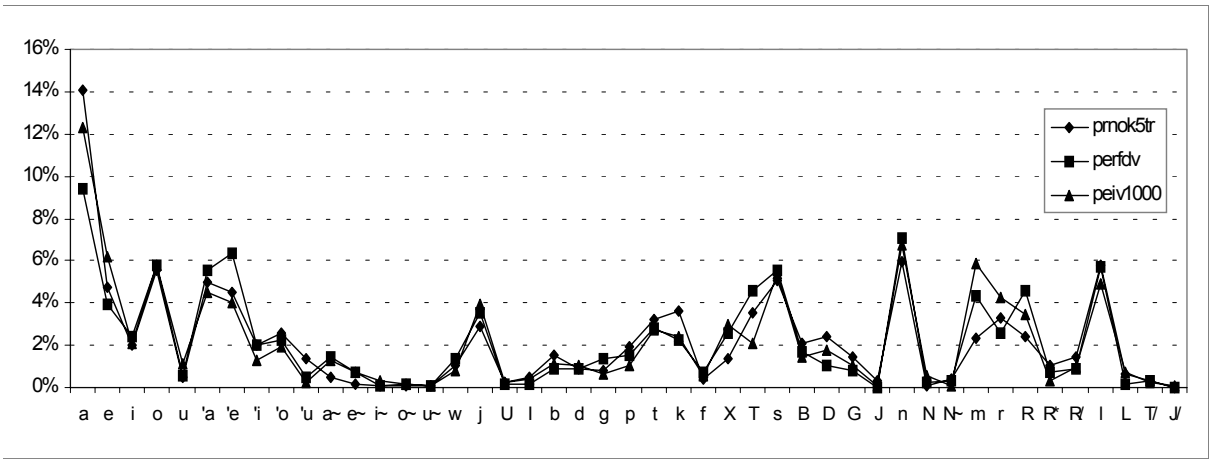

Figura D-3: Distribución de las ocurrencias de las unidades de `alf45` en las bases de datos de VESTEL

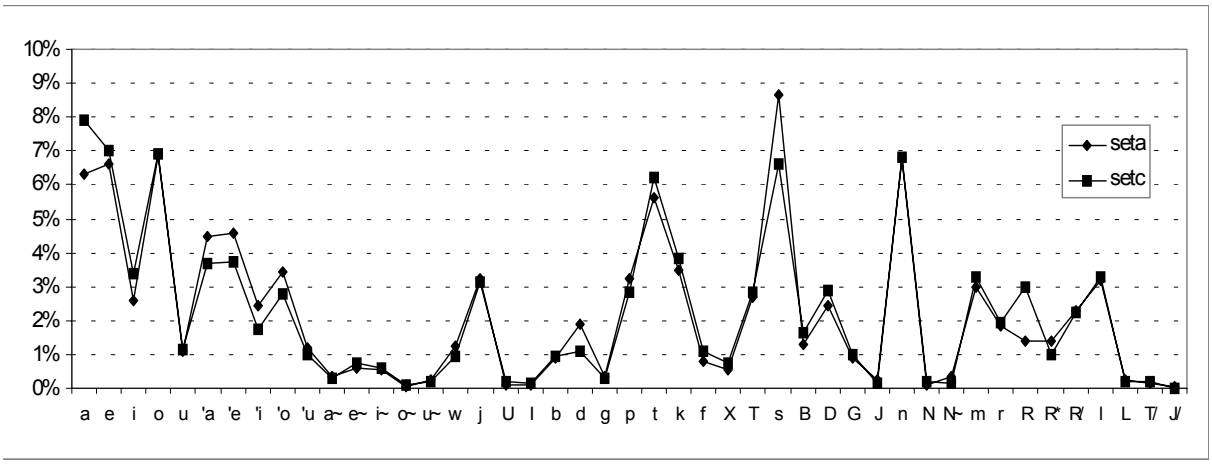

Figura D-4: Distribución de las ocurrencias de las unidades de `alf45` en los subconjuntos de POLYGLOT

que puede considerarse como razonable, mientras que todas las mostradas para POLYGLOT tendrán un entrenamiento muy poco robusto.

Tabla D-4: Número de unidades de `alf45` disponibles en las listas de entrenamiento con posibles problemas de número de repeticiones (posible entrenamiento deficiente)

ALF45	Símbolo	prnok5tr	set-a
11	e~	48	
12	i~	21	
13	o~	24	**2**
14	u~	17	
17	U		**3**
18	I		**3**
34	N	33	**3**
43	T/		5
44	J/	14	**2**

D.2.3 Alfabeto: `alf23`

Procede de alf51 y es el resultado de eliminar los problemas de robustez de entrenamiento en algunas de las unidades sobre las tareas de POLYGLOT. Se trataba de diseñar un repertorio mínimo que mantuviera en lo posible la integridad acústica necesaria, pudiendo ser igualmente utilizable en sistemas en los que no hiciera falta un refinamiento acústico elevado (módulos de preselección, por ejemplo).

D.2.3.1 Contenido

Es el mostrado en la Tabla D-5.

Tabla D-5: Contenido del alfabeto `alf23`. Total: 23 unidades

Código	Símbolo	Descripción	Ejemplo
0	a	Vocales de todo tipo (semivocales y semiconsonantes incluidas)	cas**a**
1	e		esp**e**ra
2	i		hac**i**a acierto arnaiz
3	o		c**o**lega
4	u		**u**sted acuerdo araujo
5	b	Oclusivas (B, D y G fricativas integradas en b, d, g oclusivas)	**b**odega abogado
6	d		**d**ado abogado
7	g		**g**ato abogado
9	p		**p**etaca
9	t		pe**t**aca
10	k		peta**c**a
11	f	Fricativas	**f**igura
12	X		e**j**emplo
13	T		a**c**ierto
14	s		**s**onido
15	n	Nasales (N integrada en n)	a**n**a bengoa
16	N~		a**ñ**o
17	m		a**m**o
18	r	Vibrantes	ca**r**o
19	R		ca**rr**o is**r**ael césar crespo

Tabla D-5: Contenido del alfabeto `alf23`. Total: 23 unidades

Código	Símbolo	Descripción	Ejemplo
20	l	Laterales (J/ africada y J fricativa integradas en L)	ho**l**a
21	L		o**ll**a amaya yuntas
22	T/	Africadas	o**ch**o

En la misma tabla se indican las integraciones realizadas.

D.2.3.2 Estadísticas de ocurrencias

En la Figura D-5 y la Figura D-6 se incluye la distribución estadística de las ocurrencias de cada una de las unidades del alfabeto `alf23` en las bases de datos disponibles en VESTEL y POLYGLOT, respectivamente.

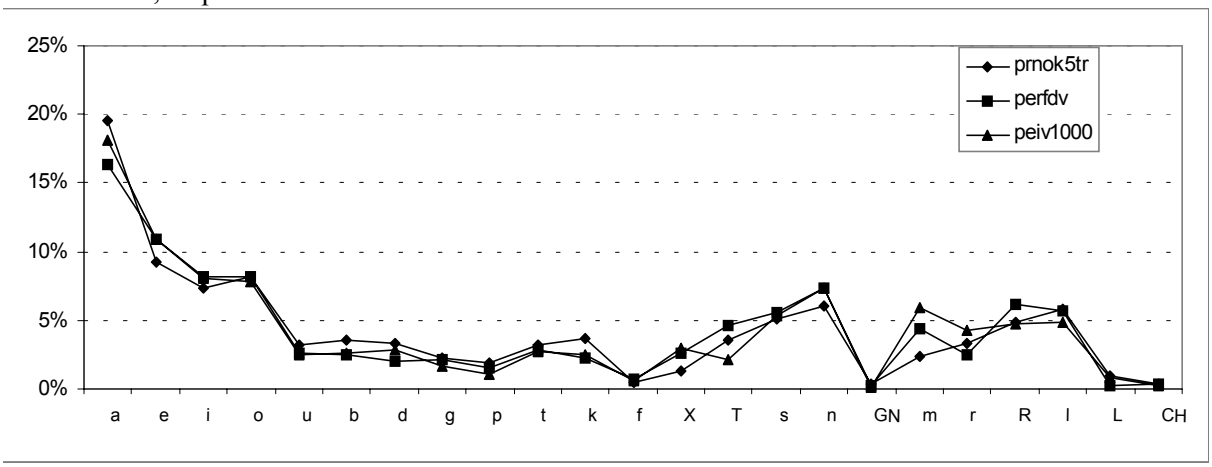

Figura D-5: Distribución de las ocurrencias de las unidades de `alf23` en las bases de datos de VESTEL

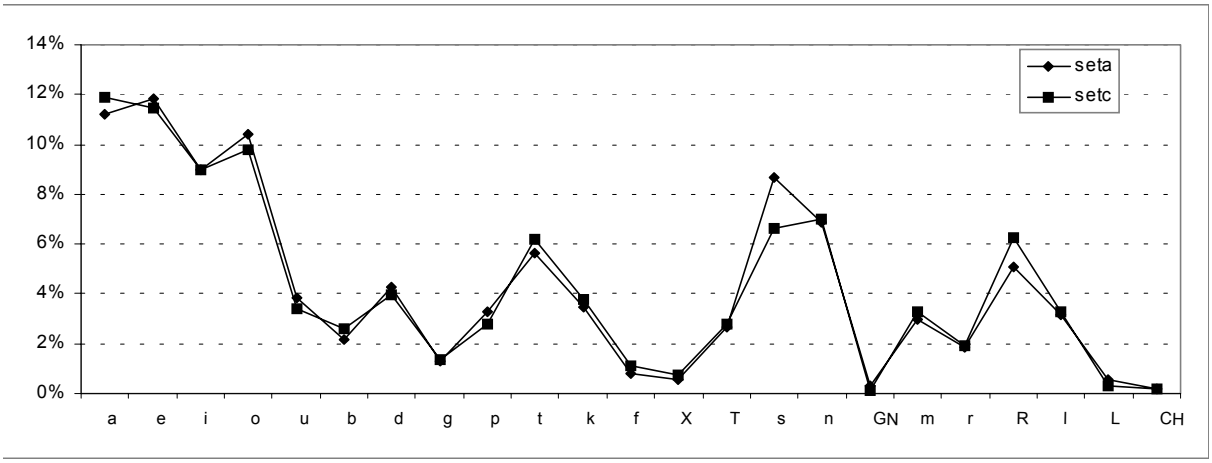

Figura D-6: Distribución de las ocurrencias de las unidades de `alf25` en los subconjuntos de POLYGLOT

Así, en lo que respecta al número global de ocurrencias en las listas de entrenamiento, incluimos en la Tabla D-6 los valores correspondientes a las unidades más escasas[1]. Como puede

observarse, las unidades más problemáticas en POLYGLOT cuentan ahora con un número de ocurrencias mínimo de 5, suficiente para nuestros propósitos.

Tabla D-6: Número de unidades de `alf25` disponibles en las listas de entrenamiento con posibles problemas de número de repeticiones (posible entrenamiento deficiente)

ALF25	Símbolo	prnok5tr	set-a
11	f	159	
16	GN	144	10
22	T/	115	5

D.2.4 Alfabeto: `alf33`

Procede de alf51 y es el resultado de agrupar unidades con contenido acústico similar en un punto intermedio entre la exhaustividad de alf51 y alf45, y la simplicidad de alf25.

D.2.4.1 Contenido

Es el mostrado en la Tabla D-7.

Tabla D-7: Contenido del alfabeto `alf33`. Total: 33 unidades

Código	Símbolo	Descripción	Ejemplo
0	a	Vocales de todo tipo	cas**a**
1	e		esp**e**ra
2	i		hac**i**a
3	o		c**o**lega
4	u		**u**sted
5	w	Semiconsonantes	ac**u**erdo
6	j		ac**i**erto
7	U	Semivocales	arau**j**o
8	I		arna**i**z
9	b	Oclusivas	**b**odega
10	d		**d**ado
11	g		**g**ato
12	p		**p**etaca
13	t		pe**t**aca
14	k		peta**c**a

1. El criterio de definición de escasez ha sido el considerar aquellas unidades cuyo número de ocurrencias es inferior al 10% del porcentaje que correspondería a una distribución uniforme (mismo número de ocurrencias para todas las posibles). Por ejemplo, en PRNOK disponemos de 37325 unidades pertenecientes a `alf51` (51 unidades), con lo que marcarán como insuficientes (a priori) aquellas que no superen las 37325/51/10 = 78 unidades.

Tabla D-7: Contenido del alfabeto `alf33`. Total: 33 unidades

Código	Símbolo	Descripción	Ejemplo
15	f	Fricativas	**f**igura
16	X		e**j**emplo
17	T		a**c**ierto
18	s		**s**onido
19	B		a**b**ogado
20	D		aboga**d**o
21	G		abo**g**ado
22	J		ama**y**a
23	n	Nasales	a**n**a
24	N		be**n**goa
25	N~		a**ñ**o
26	m		a**m**o
27	r	Vibrantes	ca**r**o
28	R		ca**rr**o is**r**ael césar crespo
29	l	Laterales	ho**l**a
30	L		o**ll**a
31	T/	Africadas	o**ch**o
32	J/		**y**untas

Como puede observarse, al compararlo con la Tabla D-1, se han unificado las vocales nasalizadas, ya estén acentuadas o no, al igual que todas las vibrantes, unificándola con la R doble convencional.

D.2.4.2 Estadísticas de ocurrencias

En la Figura D-7 y la Figura D-8 se incluye la distribución estadística de las ocurrencias de cada una de las unidades del alfabeto `alf33` en las bases de datos disponibles en VESTEL y POLYGLOT, respectivamente.

Así, en lo que respecta al número global de ocurrencias en las listas de entrenamiento, incluimos en la Tabla D-8 los valores correspondientes a las unidades más escasas.

Tabla D-8: Número de unidades de alf33 disponibles en las listas de entrenamiento con posibles problemas de número de repeticiones (posible entrenamiento deficiente)

ALF33	Símbolo	prnok5tr	set-a
17	U	85	**3**
18	I		**3**
34	N	33	**3**
43	T/		5
44	J/	14	**2**

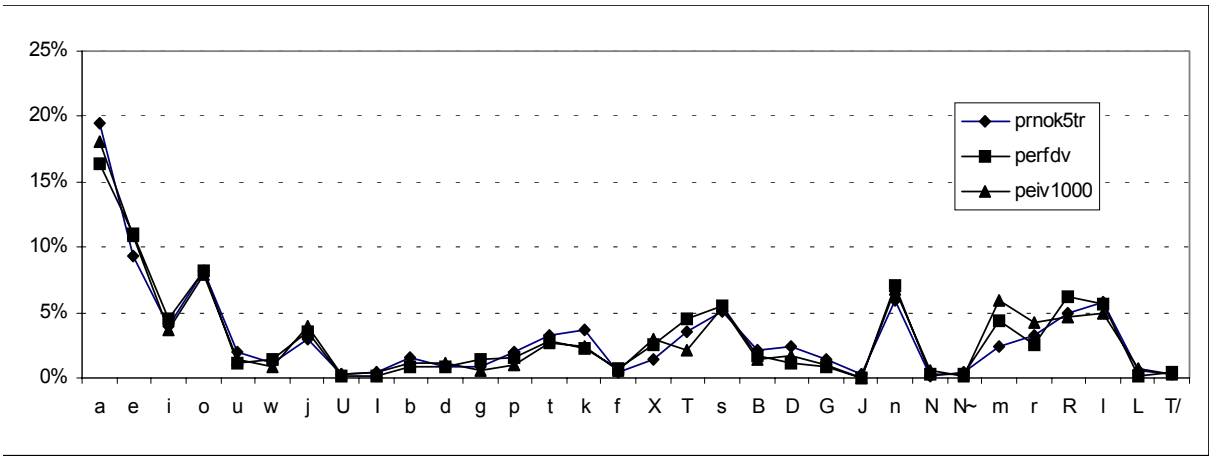

Figura D-7: Distribución de las ocurrencias de las unidades de `alf33` en las bases de datos de VESTEL

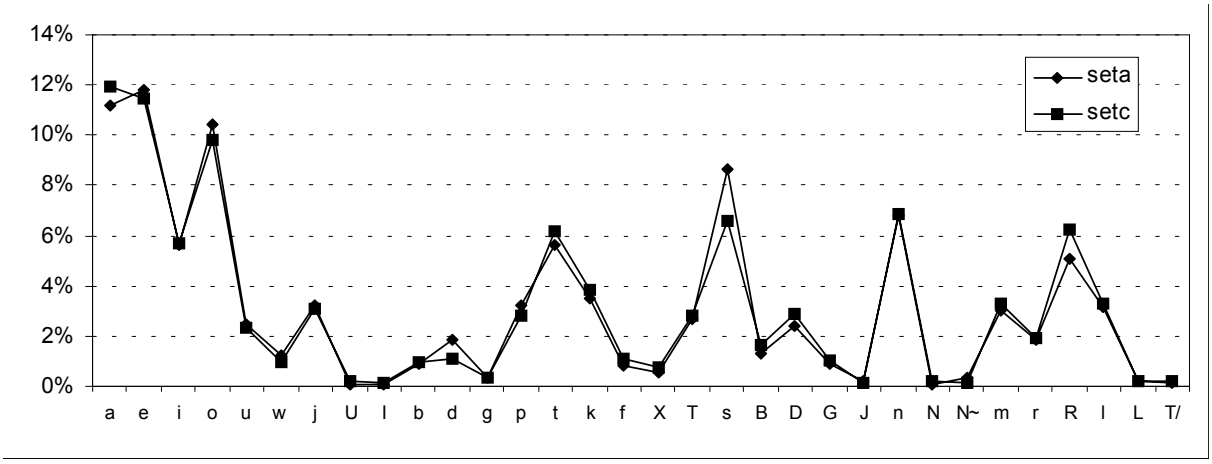

Figura D-8: Distribución de las ocurrencias de las unidades de `alf33` en los subconjuntos de POLYGLOT

D.3 Alfabetos automáticos

Incluimos en este apartado únicamente el alfabeto generado automáticamente a partir del de 51 unidades y compuesto por 23 unidades, a modo de ejemplo del tipo de agrupaciones que pueden esperarse.

D.3.1 Alfabeto: `alf_cl23`

D.3.1.1 Contenido

No vamos a describir en este caso el contenido del alfabeto, sino los símbolos agrupados resultantes. Es el mostrado en la Tabla D-1.

Tabla D-9: Contenido del alfabeto `alf_cl23`. Total: 23 unidades

Código	Símbolo	Unidades originales (de `alf51`)
0	a	a

Tabla D-9: Contenido del alfabeto `alf_cl23`. Total: 23 unidades

Código	Símbolo	Unidades originales (de `alf51`)
1	e	e
2	i	i ’i I
3	o	o
4	u	u U ’u w
5	’a	’a a~ ’a~
6	’e	’e
7	’o	’o
8	e~	e~ i~ ’i~ J/ N~ J L
9	o~	o~ u~ ’e~ ’u~ R_ ’o~ N G B D
10	j	j
11	b	b d g
12	p	p k
13	t	t
14	f	f T/ T
15	X	X
16	s	s
17	n	n
18	m	m
19	r	r R/
20	R	R
21	R*	R*
22	l	l

En las agrupaciones vistas pueden observarse algunas coherentes desde el punto de vista acústico-fonético y otras más sorprendentes. Del análisis de estas agrupaciones se desprende que se deben fundamentalmente a la mayor presencia de contextos en los que aparecen ambas unidades simultáneamente, unido a una escasa aparición de la unidad aglomerada.

D.3.1.2 Estadísticas de ocurrencias

En la Figura D-9 se incluye la distribución estadística de las ocurrencias de cada una de las unidades del alfabeto `alf_cl23` en las bases de datos disponibles en VESTEL, respectivamente. Como puede observarse, dicha distribución es, de nuevo, razonablemente homogénea en los conjuntos de entrenamiento y reconocimiento.

Igualmente, el efecto de escasez de ocurrencias para algunas unidades que veíamos en alfabetos más complejos como `alf51` e incluso en los manuales como `alf25` se ve aliviado, ya que el proceso de agrupación también tiene en cuenta este factor. Así, en lo que respecta al número global de ocurrencias en las listas de entrenamiento, incluimos en la Tabla D-10 los valores correspondientes a las unidades más escasas que, en este caso, superan ampliamente el valor utilizado en apartados anteriores referido como umbral de escasez[1]. Con esto se muestra cómo los criterios de agrupación utilizados producen un efecto beneficioso adicional: balanceo más razonable de los ejemplos de entrenamiento disponibles, lo que constituye una explicación más del buen funcionamiento obtenido con los modelos agrupados automáticamente.

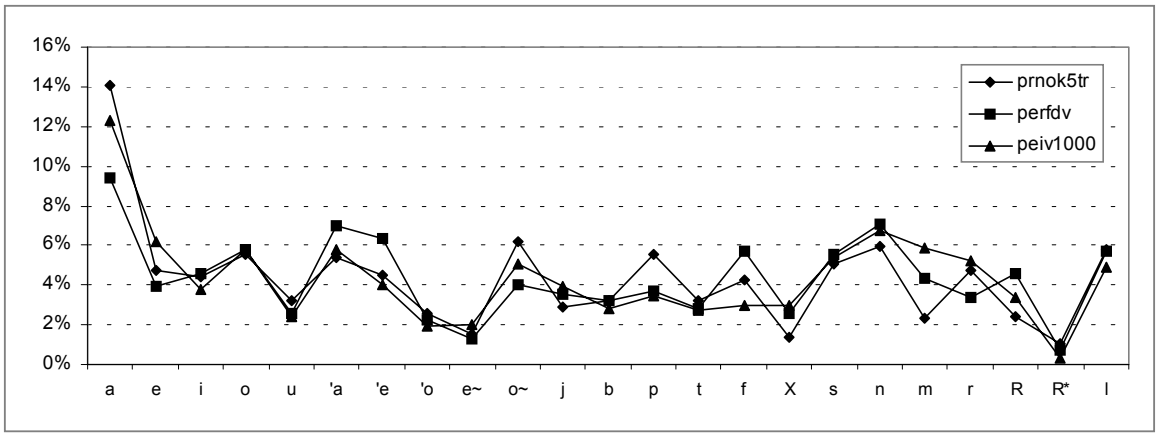

Figura D-9: Distribución de las ocurrencias de las unidades de `alf_cl23` en las bases de datos de VESTEL

Tabla D-10: Número de unidades de `alf_cl23` disponibles en las listas de entrenamiento con posibles problemas de número de repeticiones (posible entrenamiento deficiente)

alf_cl_23	Símbolo	prnok5tr
21	R*	385
15	X	501
8	e~	556
18	m	880
20	R	907
7	'o	953

1. El criterio de definición de *umbral de escasez* fue el considerar aquellas unidades cuyo número de ocurrencias es inferior al 10% del porcentaje que correspondería a una distribución uniforme (mismo número de ocurrencias para todas las posibles).

E Validación estadística

Aunque existen diversos métodos de validación estadística de resultados, el que se ha utilizado en esta tesis es el cálculo de bandas de probabilidad, puesto que es más restrictivo que, por ejemplo, el test de McNemar [Hunt90], siguiendo con las ideas aplicadas en este sentido en [Ferreiros99].

Para el cálculo de bandas de probabilidad utilizamos la siguiente fórmula [Weiss93]:

$$\frac{banda}{2} = 1,96 \cdot \sqrt{\frac{p(100-p)}{n}} \qquad \text{(EQ 1)}$$

donde **p** es el porcentaje de reconocimiento obtenido comparando con **n** palabras de hipótesis y el porcentaje real está con un 95 % de confianza en el intervalo (p-banda/2, p+banda/2). Una vez obtenidas estas bandas, utilizaremos el criterio de que las bandas de dos sistemas no se solapen en absoluto para que podamos asegurar que su comportamiento es definitivamente diferente, aunque como se comenta en el encuadre científico tecnológico de esta tesis, puede suceder que dicha validación estadística no se cumpla significativamente, pero se verifiquen mejoras apreciables en todos los casos de aplicación de una técnica determinada. Así, dichas técnicas "son ideas que se proponen con la cautela de no haber podido demostrar significativamente su validez, pero que no queremos abandonar a la espera de que sean útiles en los sistemas de reconocimiento y que queden validadas en futuras experimentaciones en otras aplicaciones o con más datos" [Ferreiros96].

Bibliografía

[1] [Adda99]
M. Adda-Decker, L. Lamel
Pronunciation variants across system configuration, language and speaking style
Speech Communication 29 (1999), pp. 225-246. 1999.

[2] [Aktas91]
A. Aktas y K. Zünkler.
Speaker Independent Continuous HMM-Based Recognition of Isolated Words on a Real-Time Multi-DSP System
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1991. pp. 1345-1348. 1991

[3] [Antoniol90]
G. Antoniol, F. Brugnara y D. Giuliani.
Admisible Strategies for Acoustic Matching with a Large Vocabulary.
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1990. pp. 589-592, 1990.

[4] [Aubert94]
X. Aubert, C. Dugast, H. Ney y V. Steinbiss
Large Vocabulary Continuous Speech Recognition of Wall Street Journal Data
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1994, vol. 2, pp. 129-132. 1994

[5] [Aubert95]
X. Aubert y Christian Dugast
Improved Acoustic-phonetic Modelling in PHILIPS Dictation System by Handling Liasions and Multiple Pronunciations
Proc. of the European Conference on Speech Communication and Technology (Eurospeech), 1995, pp. 767-770. 1995

[6] [Bahl83]
L.R. Bahl, F. Jelinek y R.L. Mercer
A Maximum Likelyhood Approach to Continuous Speech Recognition
IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. PAMI-5, nº 2, pp. 179-190. 1983

[7] [Bahl85]
L. R. Bahl, P. F. Brown y P. V. de Souza
A Fast Algorithm for Deleted Interpolation
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1985. pp. 1209-1212, 1985.

[8] [Bahl86]
L.R. Bahl, P.F. Brown, P.V. de Souza y R.L. Mercer
Maximum Mutual Information Estimation of Hidden Markov Model Parameters for Speech Recognition
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1986, pp. 49-52. 1986

[9] [Bahl89]
L. R. Bahl et al
Large Vocabulary Natural Language Continuous Speech Recognition
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1989, pp.. 465-467. 1989.

[10] [Bahl92]
L.R. Bahl, P.V. de Souza, Gopalakrishnan, D. Nahamoo y M. Picheny
A Fast Match for Continuous Speech Recognition Using Allophonic Models
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1992, vol 1, pp. 17-20. 1989

[11] [Bahl93a]
L.R. Bahl, P.F. Brown, P.V. de Souza, y R.L. Mercer
E*stimating Hidden Markf Model Parameters So As to Maximize Speech Recognition Accuracy*
IEEE Transactions on Speech and Audio Processing, 1993, vol 1 nº 1, pp 77-83. 1993.

[12] [Bahl93b]
L.R. Bahl, P.F. Brown, P.V. de Souza, R.L. Mercer y M.A. Picheny.
A Method for the Construction of Acoustic Markov Models for Words
IEEE Transactions on Speech and Audio Processing, 1993, vol 1, nº 4, pp. 443-452. 1993.

[13] [Bahl93c]
L.R. Bahl, S.V. de Gennaro, P.S. Gopalakrishnan y R.L. Mercer
Fast Approximate Acoustic Match for Large Vocabulary Speech Recognition
IEEE Transactions on Speech and Audio Processing, 1993, vol 1, pp. 59-67. 1993

[14] [Baker75]
J. Baker
The DRAGON System - An overview
IEEE Transactions on Acoustics, Speech and Signal Processing, vol 23, no 1, pp. 24-29. 1975.

[15] [Bakis82]
R. Bakis.
"Continuous Speech Word Recognition via centisecond acoustic states".
Proceedings of the Acoustical Society of America, Washington, DC, abril 1982.

[16] [Baker90]
J. Baker.
Large Vocabulary Speaker-Adaptive Continuous Speech Recognition Research Overview at Dragon Systems
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1990. pp. 29-31. 1990.

[17] [Bellman57]
R. Bellman
Dynamic Programming
Princeton University Press, 1957

[18] [Beyerlein94]
P. Beyerlein
Fast Log-Likelihood Computation for Mixture Densities in a High Dimensional Feature Space
Proc. of the International Conference on Spoken Language Processing (ICSLP), 1994, Yokohama (Japón), pp. 263-266. 1994.

[19] [Bezie93]
O. Bezie, O. y P. Lockwood
Beam Search and Partial Traceback in the Frame Synchronous Two Level Algorithm
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol 2, 1993, pp. 511-514. 1993.

[20] [Billi89]
R. Billi, G. Arman, D. Cericola, G. Massia, M. J. Mollo, F. Tafini, G. Varese y V. Vittorelli.
A PC-Based Very Large Vocabulary Isolated Word Speech Recognition System
Proc. of the European Conference on Speech Communication and Technology (Eurospeech),

1989. pp. 157-160. 1989.

[21] [Bishop95]
C.M. Bishop
Neural Networks for Pattern Recognition, Oxford: Oxford University Press. 1995

[22] [Bocchieri93]
E. Bocchieri
Vector Quantization for Efficient Computation of Continuous Density Likelihoods
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1993, vol II, pp 692-695. 1993.

[23] [Bourlard93]
H. Bourlard y N. Morgan
Connectionist Speech Recognition - A hybrid Approach
Kluwer Academic, 1993

[24] [Bourlard96]
H. Bourlard, H. Hermansky y N. Morgan
Towards increasing speech recognition error rates
Speech Communication, vol 18, nº 3, pp 205-232. 1996.

[25] [Bridle79]
J. S. Bridle y M. D. Brown
Connected word recognition using whole word templates
Proceedings of the Institute of Acoustics. Noviembre 1979, pp. 25-28, 1979.

[26] [Bridle82]
J. S. Bridle, M. D. Brown y Chamberlein.
An Algorithm for Connected Word Recognition
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1982, pp. 899-902, 1982.

[27] [Brow82]
P. F. Brown, J. C. Spohrer, P. H. Hochschild y J. K. Baker.
Parcial Traceback and Dynamic Programming
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1982, pp. 1629-1632

[28] [Buttafava90]
P. Buttafava, R. Billi, W. Digiampietro, G. Massica, V. Vittorelli.
Architecture and Implementation of the Olivetti PC-Based Very Large Vocabulary Isolated Word Speech Recognition System
Proc. of the European Conference on Speech Communication and Technology (Eurospeech), 1990.

[29] [Cananesio92]
F. Canavesio, G. Castagneri, G. di Fabbrizio y F. di Senia
Comparison between two methodologies of testing isolated word speech recognizers
Proc. of the International Conference on Spoken Language Processing (ICSLP), 1992, Banff, pp. 1375-1378. 1992.

[30] [Cerf92]
H. Cerf-Danon, S. DeGennaro, M. Ferreti, J. González y E. Keppel
1.0 Tangora - A Large Vocabulary Speech Recognition System for Five Languages
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1992. pp. 183-191, 1992.

[31] [Chengalvarayan97]
R. Chengalvarayan y L. Deng

Use of Generalized Dynamic Feature Parameters for Speech Recognition
IEEE Transactions on Acoustics, Speech and Signal Processing, vol 5, nº 3, pp 232-242, Mayo 1997

[32] [Chiang94]
T.H. Chiang, Y.C. Lin, K.Y. Su
A Study of Aplying Adaptive Learning to a Multi-Module System
Proc. of the International Conference on Spoken Language Processing (ICSLP), 1994, pp. 589-592. Yokohama (Japón)

[33] [Chiang96]
T.H. Chiang, Y.C. Lin y K.Y. Su
On Jointly Learning the Parameters in a Character Synchronous Integrated Speech and Language Model
IEEE Transactions on Speech and Audio Processing, 1996, vol 4, nº 3, pp 167

[34] [Cifuentes91]
S. Cifuentes.
Reconocimiento de Cadenas de Dígitos
Proyecto Fin de Carrera. Departamento de Ingeniería Electrónica, ETSITM-UPM. Madrid, 1991

[35] [Colás99]
José Colás Pasamontes
Estrategias de incorporación de conocimiento sintáctico y semántico en sistemas de comprensión de habla continua en castellano
Tesis Doctoral. ETSIT UPM. Madrid 199p

[36] [Cole95]
R. Cole, ed. J. Mariani, H. Uszkoreit, A. Zaenen, y V. Zue
Survey of the State of the Art in Human Language Technology
http://www.cse.ogi.edu/CSLU/HLTsurvey/HLTsurvey.html, (en línea) 1995

[37] [Córdoba95]
Ricardo de Córdoba Herralde
Sistemas de Reconocimiento de Habla Continua y Aislada: Comparación y Optimización de los Sistemas de Modelado y de los Parámetros del Habla
Tesis Doctoral. ETSIT UPM. Madrid 1995

[38] [Deng88]
L. Deng, M. Lenning, V. N. Gupta, P. Mermelstein
Modeling Acoustic-Phonetic Detail in an HMM based Large Vocabulary Speech Recognizer
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1988. pp. 509-512, 1988.

[39] [Deng90]
L. Deng, V. Gupta, M. Lenning, P. Kenny y P. Mermelstein.
Acoustic Recognition Component of an 86000-Word Speech Recognizer
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1990. pp. 741-744.

[40] [Elvira97]
J. M. Elvira, J.C. Torrecilla y J. Caminero
Creating User Defined New Vocabularies for Voice Dialing
Proc. of the European Conference on Speech Communication and Technology (Eurospeech), 1997, Volume 5 pp. 2463-2466. 1997.

[41] [Enríquez00]
E. Enríquez
Notas de variantes dialectales o locales del español peninsular de posible interés para sistemas

de análisis de voz
Informe interno Grupo de Tecnología del Habla. GTH-2-00. 2000

[42] [Ferreiros96]
Javier Ferreiros
Aportación a los métodos de entrenamiento de Modelos de Markov para reconocimiento de habla continua
Tesis Doctoral. ETSIT UPM. Madrid 1996

[43] [Ferreiros99]
J.Ferreiros y J.M.Pardo
Improving continuous speech recognition in Spanish by phone-class semicontinuous HMMs with pausing and multiple pronunciations
Speech Communication, 29, pp. 65-76, Septiembre 1999.

[44] [Fetter96]
P. Fetter, F. Dandurand y P. Regai-Brietzmann
Word Graph Rescoring Using Confidence Measures
Proc. of the International Conference on Spoken Language Processing (ICSLP), 1996, Philadephia, pp. 10-13. 1996.

[45] [Fissore89]
L. Fissore, P. Laface, G. Micca y R. Pieraccini
Lexical Acces to Large Vocabularies for Speech Recognition
IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 17, nº 8, pp. 1197-1213, agosto 1989.

[46] [Fissore91]
Fissore, L., E. Giachin, P. Laface y G. Micca
Selection of Speech Units fos a Speaker-Independent CSR Task
Proc. of the European Conference on Speech Communication and Technology (Eurospeech), 1991, vol. 3, pp. 1389-1392. 1991.

[47] [Gaviña00]
David Gaviña Barroso
Distintas alternativas de compartición de parámetros en modelos HMM continuos en un sistema de reconocimiento de habla aislada
Proyecto fin de carrera. ETSIT-UPM. 2000

[48] [Gibbon98]
D. Gibbon, R. Moore y R. Winski, Editores
Handbook of Standards and Resources for Spoken Language Systems
Volumen III. Ed. Mouton de Gruyter, Nueva York, 1998

[49] [Gillick89]
L. Gillick y S.J. Cox
Some statistical Issues in the Comparison of Speech Recognition Algorithms
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1989. pp 532-535. 1989

[50] [Gopalakrishanan95]
P.S. Gopalakrishnan, L.R. Bahl y L.R., Mercer
A Tree Search Strategy for Large-Vocabulary Continuous Speech Recognition
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1995, vol 1, pp.. 572-575. 1995.

[51] [Gravier97]
G. Gravier, F. Yvon, G. Etorre, G. Chollet, ENST
Directory Name Retrieval Using HMM Modeling and Robust Lexical Access

Automatic Speech Recognition and Understanding Workshop, December 14-17, 1997. Santa Barbara, California, USA. 1997.

[52] [Gray84]
R. Gray
Vector Quantization
IEEE Acoustics, Speech and Signal Processing Magazine, vol. 1, nº 2, pp. 4-20, abril 1984.

[53] [Haeb94]
R. Haeb-Umbach, D. Geller y H. Ney
Improvements in Connected Digit Recognition using Linear Discriminant Analysis and Mixture Densities
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1994, vol2 pp 239-242. 1994.

[54] [Hasan90]
H. Hasan
Reconocimiento de 1000 Palabras Independiente del Locutor Mediante Modelos Ocultos de Markov
Tesis Doctoral. ETSITM-UPM, julio 1990.

[55] [Hassoun95]
M. H. Hassoun
Fundamentals of Artificial Neural networks, MIT Press, 1995.

[56] [Hermansky94]
H. Hermansky, N. Morgan, A. Bayya y P. Kohn.
Rasta-PLP Speech Analysis Technique
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1992, pp. 121-124. 1992.

[57] [Hochberg94]
M. Hochberg, T. Robinson y S. Renals
Large Vocabulary Continuous Speech Recognition using a Hybrid Connectionist HMM System
Proc. of the International Conference on Spoken Language Processing (ICSLP), 1994, pp. 1499-1502. 1994.

[58] [Holter98]
T. Holter
Maximul Likelihood Modelling of Pronunciation in Automatic Speech Recognition
PhD. Thesis. Department of Telecommunications. Signal Processing Group. Norwegian University of Science and Technology, 1998.

[59] [Hon89]
H.W. Hon , K.F. Lee, R. Weide.
Towards Speech Recognition Without Vocabulary- Specific Training
Proc. of the European Conference on Speech Communication and Technology (Eurospeech), 1989, pp. 481-484. 1989.

[60] [Hon90]
H.W. Hon y K.F. Lee
On Vocabulary-Independent Speech Modeling
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1990, pp. 725-728. 1990.

[61] [Hood91]
M. Hood.
Lexical Access in a Speech Understandig and Dialogue System
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing

(ICASSP) 1991. pp. 490-493, 1991.

[62] [Hoste00]
Véronique Hoste, Steven Gillis y Walter Daelemans.
A Rule Induction Approach to Modeling Regional Pronunciation Variation
Proc. of COLING 2000, pp. 327-333. Saarbrücken, Germany. San Francisco: Morgan Kaufman Publishers, 2000.

[63] [Huang89]
X.D. Huang, H.W. Hon y K.F. Lee
Large-Vocabulary speaker independent CSR with semi-continuous HMMs
Proc. of the European Conference on Speech Communication and Technology (Eurospeech), 1989 pp. 163-166. 1989.

[64] [Huang90a]
B.H. Juang y L. R. Rabiner.
The segmental k-means algorithm for estimating parameters of hidden Markov models
IEEE Transactions on Speech and Audio Processing, Vol. 38, pp.1639--1641, 1990.

[65] [Huang90b]
X.D. Huang, Y. Ariki y M.A. Jack
Hidden Markov Models for speech recognition
Edinburgh University Press, 1990.

[66] [Huang93]
X.D. Huang, H.W. Hon, M.Y. Hwang, K.F. Lee.
A comparative study of discrete, semicontinuous and continuous HMMs
Computer Speech and Language, nº 7, pp. 359-368, 1993

[67] [Hwang93a]
M.Y. Hwang, X.D. Alleva y X.D. Huang.
Shared-distribution HMM for Speech Recogition
IEEE Transactions on Speech and Audio Processing, vol. 1, nE 4, pp. 414-420. 1993

[68] [Hwang93b]
M.Y. Hwang, X.D. Alleva y X.D. Huang.
Senones, Multi-Pass Search and Unified Stochastic Modelling in SPHINX-II
Proc. of the European Conference on Speech Communication and Technology (Eurospeech), 1993, vol. 3, pp. 2143-2146. 1993.

[69] [Hunt89]
M. Hunt y C. Lefebvre
A Comparison of Several Acoustic Representations for Speech Recognition with Degraded and Undegraded Speech
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1989, pp 262-265. 1989.

[70] [Iwasaki97]
T. Iwasaki y Y. Abe
A Memory Management Method for a Large Word Network
Proc. of the European Conference on Speech Communication and Technology (Eurospeech), 1997, pp 171-174. 1997.

[71] [Jelinek76]
F. Jelinek
Continuous Speech Recognition by Statistical Methods
Procceedings of the IEEE, vol 64, no 4, pp. 532-555, 1976

[72] [Jelinek91]
F. Jelinek

Up from Trigrams! - the Struggle for Improved Language Models
Proc. of the European Conference on Speech Communication and Technology (Eurospeech) 1991, pp. 1037-1040. 1991.

[73] [Jones94]
G. Jones
Application of Linguistic Models to Continuous Speech Recognition
Tesis Doctoral. Universidad de Bristol. 1994.

[74] [Juang97]
B.-H. Juang, W. Chou y C.-H. Lee
Minimum Classification Error Rate Methods for Speech Recognition
IEEE Transactions on Speech and Audio Processing, 1997, vol 5, nº 3, pp 257-265. 1997.

[75] [Kenny92]
P. Kenny, S. Parthasarathy, V. N. Gupta, M. Lenning, P. Mermelstein y D. O'Shaughnessy.
Energy, Duration and Markov Models
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1992. pp. 655-658, 1992.

[76] [Knill96]
L.M. Knill, M.J.F. Gales, S. Young
Use of Gaussian Selection in Large Vocabulary CSR using HMMs
Proc. of the International Conference on Spoken Language Processing (ICSLP), 1996, pp 470-473. 1996.

[77] [Labute95]
Labute, P., Kenny, P., Hollan, R., Lenning. M. O'Shaughnessy, D.
A New Fast Match for Very Large Vocabulary Speech Recognition
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1995, vol 2, pp. 656-659. 1995.

[78] [Laface94]
P. Laface, L. Fissore, yF. Ravera,
Automatic Generation of Words Toward Flexible Vocabulary Isolated Word Recognition
Proc. of the International Conference on Spoken Language Processing (ICSLP), 1994, Yokohama, pp. 2215-2218. 1994.

[79] [Lang90]
K.J. Lang, A. Waibel y G.E. Hinton
A time-delay Neural Network Architecture for Isolated Word Recognition
Neural Networks, 3(1), pp 23-43. 1990.

[80] [Leandro94]
Manuel Antonio Leandro Reguillo
Técnicas Eficientes para Reconocimiento de Habla en Español
Tesis Doctoral. ETSIT UPM. Madrid 1994

[81] [Lee88]
K. F. Lee.
Large-Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System
Tesis Doctoral. Carnegie Mellon University. 1988.

[82] [Lee89a]
K.F. Lee
Hidden Markov Models: Past, Present and Future
Proc. of the European Conference on Speech Communication and Technology (Eurospeech), 1989, pp. 148-155. 1989.

[83] [Lee89b]

K. F. Lee.
Automatic Speech Recognition - the development of the SPHINX System
Kluwer Academic Publishers, 1989.

[84] [Lee90]
C.H. Lee, L.R. Rabiner, R. Pieraccini y J. Wilpon.
Acoustic modelling of subword units for speech recognition
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1990, pp. 721-724. 1990.

[85] [Lee97]
C. Z. Lee, D. O'Shaughnessy, INRS-Telecommunications
Techniques to Achieve Fast Lexical Access
Automatic Speech Recognition and Understanding Workshop, December 14-17, 1997. Santa Barbara, California, USA. 1997.

[86] [Levinson86]
S. Levinson.
Continuously Variable Duration HMMs for Speech Analysis
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1986. pp. 1241-1244, 1986.

[87] [Levin95]
E. Levin and R. Pieraccini.
CHRONUS - The Next Generation.
In Proceedings of ARPA Workshop on Human Language Technology, January 1995.

[88] [Li95]
Z. Li, P. Kenny y D. O'Shaughnessy
Searching with a Transcription Graph
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1995, vol 1, pp 564-567. 1995.

[89] [Lippmann89]
R.P. Lippmann.
Pattern classification using neural networks
IEEE Communication Magazine 27, 47-64. 1989.

[90] [Ljolje92]
Andrej Ljolje and Michael D. Riley.
Optimal speech recognition using phone recognition and lexical access.
Proceedings of ICSLP, pages 313--316, Banff, Canada, October 1992.

[91] [López98]
Yolanda López
Demostracion de reconocimiento de habla continua mediante modelos semicontinuos de Markov
Proyecto fin de carrera. ETSIT-UPM. 1998.

[92] [López99]
Pedro David López Rubio
Optimización de un sistema de reconocimiento de habla aislada sobre línea telefónica
Proyecto fin de carrera. ETSIT-UPM. 1999.

[93] [Mariño00]
José B. Mariño, A. Nogueiras, P. Pachès y A. Bonafonte
The demiphone: an efficient contextual subword unit for continuous speech recognition
Speech Communication, Vol. 32, No. 3, pp. 187-197, Octubre 2000.

[94] [Masters93]
T. Masters

Practical Neural Network Recipes in C++
Academic Press, Inc., 1993

[95] [Matsuura88]
H. Matsu'ura, T. Nitta, S. Hirai, Y. Takebayashi, H. Tsuboi y H. Kanazawa.
A Large Vocabulary Word Recognition System Based on Syllable Recognition and Nonlinear Word Matching
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1988. pp. 183-186, 1988.

[96] [Menéndez94a]
X. Menéndez Pidal Sendrail
Arquitecturas Neuronales y su Integración con Algoritmos de Programación Dinámica en Tareas de Reconocimiento de Habla
Tesis Doctoral. ETSIT-UPM, 1994

[97] [Moreno90]
P. Moreno, D.B. Roe, y P. Ramesh
Rejection Techniques in continuous speech recognition using HMMs
European Signal Processing Conference, Barcelona, SP-V, 1990, pp. 1383-1396. 1990.

[98] [Moreno00]
C. Moreno Asenjo
Optimización de un sistema de reconocimiento de gran vocabulario sobre línea telefónica basado en modelos continuos de Markov
Proyecto fin de carrera. ETSIT-UPM. 2000.

[99] [Morgan91]
D. P. Morgan y C.L. Scofield
Neural Networks and Speech Processing
Kluwer Academic Publishers, 1991.

[100] [Morgan95]
N. Morgan y H. Bourlard
Continuous Speech REcognition: An introduction to the hybrid HMM/connectionist Approach
IEEE Signal Processing magazine, volume 12, number 3, pages 24-42, May 1995.

[101] [Nadeu97]
C. Nadeu, P. Pachès-Leal y Biing-Hwang Juang
Filtering the time sequences of spectral parameters for speech recognition
Speech Communication , vol 22, nº 4, Septiembre 1997, pp 315-332. 1997.

[102] [Ney84]
H. Ney
The use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition
IEEE Transactions on Acoustics, Speech and Signal Processing, 1984, vol ASSP-32, nº 2. 1984.

[103] [Ney91]
H. Ney, R. Billi.
Prototype Systems for Large Vocabulary Speech Recognition: Polyglot and Spicos
Proc. of the European Conference on Speech Communication and Technology (Eurospeech), 1991. pp 193-200. 1991.

[104] [Ney92]
H. Ney, R. Haeb-Umbach, B. Tran y M. Oerder
Improvements in Beam Search for 10000 word Continuous Speech Recognition
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1992 vol 1, pp. 5-8. 1992.

[105] [Ortmanns96]

S. Ortmanns, H. Ney y A. Eiden
Language Model Look-ahead for Large Vocabulary Speech Recognition
Proc. of the International Conference on Spoken Language Processing (ICSLP), 1996, pp. 2095-2098. 1996.

[106] [Ortmanns97a]
S. Ortmanns, T. Firzlaff y H. Ney
Fast Likelihhod Computation Methods For Continuous Mixture Densities In Large Vocabulary Speech Recognition
Proc. of the European Conference on Speech Communication and Technology (Eurospeech), 1997, Vol I, pp 139-142. 1997.

[107] [Ortmans97b]
S. Ortmans, A. Eiden, H. Ney y N. Coenen
Look Ahead Techniques for Fast Beam Search
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1997, pp 1783-1786. 1997.

[108] [Ortmans97c]
S. Ortmans, y H. Ney
A word graph algorithm for large vocabulary continuous speech recognition
Computer Speech and Language, 1997 vol 11, nº 1, pp 43-72. 1997.

[109] [Ostendorf96]
M. Ostendorf, V. Digalakis y O.A. Kimball
From HMM's to segment models: A unified view of stochastic modeling for speech recognition
IEEE Transactions on Speech and Audio Processing, vol 4, pp 360-378, septiembre 1996.

[110] [Pallet89]
D.S. Pallet
Benchmark Tests for DARPA Resource Management Performance Evaluation
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1989. pp 536-539. 1989.

[111] [Rabiner83]
L.R. Rabiner, S. E. Levinson y M. M Shondhi.
"On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independet, Isolated Word Recognition".
The Bell System Technical Journal, abril 1983. pp. 1075-1105, 1983.

[112] [Rabiner86]
L.R. Rabiner, B. H. Juang
An introduction to Hidden Markov Models
IEEE Acoustics, Speech and Signal Processing Magazine, enero 1986, pp 4-15, 1986.

[113] [Rabiner89a]
L. R. Rabiner.
A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition
Proceedings of the IEEE, Nº 2, pp. 257-286, febrero 1989.

[114] [Rabiner89b]
L. R. Rabiner, C. H. Lee, B. H. Juang y J. G. Wilpon.
HMM Clustering for Connected Word Recognition
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1989, pp. 405-408, 1989.

[115] [Rabiner88]
Rabiner, L.R., J.G. Wilpon y F.K. Soong.
High performance connected digit recognition using HMMs

Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1988.

[116] [Renals94]
S. Renals, N. Morgan, H. Bourlard, M. Cohen y H. Franco
Connectionist probability estimators in HMM speech REcognition
IEEE Transactions on Speech and Audio Processing, 1994, vol 12 (1), pp 161-171. 1994.

[117] [Riley91]
M. D. Riley y A. Ljolje.
Lexical Access with a Statistically Derived Phonetic Network.
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1991. pp. 585-589, 1991.

[118] [Ripley96]
B.D. Ripley
P*attern Recognition and Neural Networks*, Cambridge: Cambridge University Press. 1996.

[119] [Robinson91]
T. Robinson y F. Fallside
A Recurrent error propagation network speech recognition system
Computer Speech and Language, 5, pp. 259-274, 1991.

[120] [Robinson95]
T. Robinson, M. Hochberg y S.Renals
The use of recurrent networks in continuous speech recognition, en Automatic Speech and Speaker Recognition - Advanced Topics, Capítulo 19.
Editores: C H Lee, K K Paliwal y F.K. Soong
Kluwer Academic Publishers, 1995

[121] [Roe94]
D.B. Roe, D.B y M.D. Riley
Prediction of Word Confusabilites for Speech Recognition
Proc. of the International Conference on Spoken Language Processing (ICSLP), 1994 , pp 227-230. 1994.

[122] [Rossi94]
M. Rossi
Automatic Segmentation: Why and what segments?
Proc. of the International Conference on Spoken Language Processing (ICSLP), 1994, pp 237-240. 1994.

[123] [Sagerer96]
G. Sagerer, H. Rautenstrauch, G.A. Fink, B. Hildebrandt, A. Jusek, y F. Kummert
Incremental Generation of Word Graphs
Proc. of the International Conference on Spoken Language Processing (ICSLP), 1996, pp. 2143-2146. 1996.

[124] [Sakoe78]
H. Sakoe, y S. Chiba, S.
Dynamic programming algorithm optimization for spoken word recognition
IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-26, pp. 43-49. 1978.

[125] [SanSegundo97]
Rubén San Segundo Hernández
Optimización de un sistema de reconocimiento de habla aislada por teléfono sobre un ordenador compatible (PC)
Proyecto fin de carrera. ETSIT-UPM 1997.

[126] [Sarkkai98]

Ramesh R. Sarukkai and Dana H. Ballard
Phonetic Set Indexing for Fast Lexical Access
IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 1, Enero 1998.

[127] [Schalkoff92]
J. Schalkoff Robert
Patter recognition, statistical, structural and neural approaches
John Wiley & Sons, Inc., 1992.

[128] [Schmid93]
P. Schmid, R. Cole y M. Fanty.
Automatically Generated Word Pronunciations from Phoneme Classifier Output
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1993, vol 2. pp 223-226. 1993.

[129] [Schwartz84]
P. Schwartz, Y. Chow, S. Roucos, M. Krasner y J. Makhoul.
Improved Hidden Markov Modeling of Phonemes for Continuous Speech Recognition
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1984, paper 35.6. 1984.

[130] [Schwartz85]
R. Schwartz, Y. Chow, O. Kimball, S. Roucos, M. Krasner y J. Makhoul.
Context-Dependent Modeling for Acoustic-Phonetic Recognition of Continuous Speech
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 1985, pp. 1205-1208. 1985.

[131] [Schwartz89]
R. Schwartz, O. Kimball, F. Kubala, M. W. Feng, Y. L. Chow, C. Barry y J. Makhoul.
Robust Smoothing Methods for Discrete HMMs
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1989. pp. 548-555. 1989.

[132] [Siu97]
M.H, Siu, H. Gish y F. Richardson
Improved estimation, evaluation and applications of confidence mesaures for speech recognition
Proc. of the European Conference on Speech Communication and Technology (Eurospeech), 1997, vol. 2, pp. 831-834. 1997.

[133] [Soong86]
F. K. Soong y A. E. Rosenberg.
On the use of Instantaneous and Transitional Spectral Information in Speaker Recognition
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1986, pp. 877-880, 1986.

[134] [Soudoplatoff86]
S. Soudoplatoff
Markov Modeling of Continuous Parameters in Speech Recognition
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1986, pp. 45-48. 1986.

[135] [Steinbiss94]
V. Steinbiss, B.H. Tran, B.H. y H. Ney
Improvements in Beam Search
Proc. of the International Conference on Spoken Language Processing (ICSLP), 1994, pp. 2140-2143. 1994.

[136] [Strik99]
H. Strik y K. Cucchiarini

Modeling pronunciation variation for ASR: A survey of the literature
Speech Communication 29 (1999), pp 225-246. 1999.

[137] [Tapias94]
D. Tapias, A. Acero, J. Esteve and J.C. Torrecilla.
The VESTEL Telephone Speech Database
Proc. of the International Conference on Spoken Language Processing (ICSLP), 1994, pp. 1811-1814. 1994.

[138] [Valtchev97]
V. Valtchev, J.J. Odell, P.C. Woodland y S.J. Young
MMIE training of large vocabulary recognition systems
Speech Communication vol, 22, nº 9 pp. 303-314, Septiembre 1997.

[139] [Villarrubia96]
L. Villarrubia, L.H. Gómez, J.M. Elvira y J.C. Torrecilla
Context-dependent units for Vocabulary-independent Spanish Speech Recognition.
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 1996, pp. 451-454. 1996.

[140] [Watanabe95]
T. Watanabe
High Speed Speech Recognition Using Tree Structured Probability Density Function
Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1995, vol 1, pp. 556-559. 1995.

[141] [Weiss93]
N.A. Weiss y M.J. Hasset
Introductory Statistics
Third Edition, 1993.

[142] [Westendorf96]
C.M. Westendorf y J. Jelitto.
Learning Pronunciation Dictionary from Speech Data
Proc. of the International Conference on Spoken Language Processing (ICSLP), 1996, pp. 1045-1048. 1996.

[143] [Wooters94]
C, Wooters y A. Stolke.
Multiple-Pronunciation Lexical Modeling in a Speaker Indepent Speech Understanding System"
Proc. of the International Conference on Spoken Language Processing (ICSLP), 1994, pp. 1363-1366. 1994.

[144] [Wu94]
C, Wooters y A. Stolke.
Recognition Accuracy Methods and Measures
Proc. of the International Conference on Spoken Language Processing (ICSLP), 1994, pp. 1315-1318. 1994.

Bibliografía del autor

Se incluye en este apartado aquellas publicaciones en las que ha participado el autor de esta tesis y que tienen relación con el tema de la misma.

[1] [Córdoba01]
R. Córdoba, R. San-Segundo, J.M. Montero, J. Colás, J. Ferreiros, J. Macías-Guarasa and J.M. Pardo
An Interactive Directory Assistance Service for Spanish with Large-Vocabulary Recognition
Será publicado en Proc. of the European Conference on Speech Communication and Technology (Eurospeech) 2001. Septiembre 2001.

[2] [Ferreiros98a]
J. Ferreiros, J. Macías-Guarasa y José M. Pardo.
Introducing Multiple Pronunciations in Spanish Speech Recognition Systems
ESCA Tutorial and Research Workshop on "Modeling Pronunciation Variation for Automatic Speech Recognition", Kerkrade, Mayo 1998.

[3] [Ferreiros98b]
J. Ferreiros, J. Macías-Guarasa, A. Gallardo y L. Villarrubia.
Recent Work on a Preselection Module for a Flexible Large Vocabulary Speech Recognition System in Telephone Environment
Proc. of the International Conference on Spoken Language Processing (ICSLP). 1998. vol 2, pp. 321-324. Sidney (Australia). Noviembre 1998.

[4] [Gallardo00]
A. Gallardo, J. Ferreiros, J. Macías-Guarasa, R. de Córdoba, J. Colás and J.M. Pardo
Incorporating Multiple-HMM Acoustic Modeling in a Modular Large Vocabulary Speech Recognition System in Telephone Environment
Proc. of the International Conference on Spoken Language Processing (ICSLP) 2000, Vol. II, pp 827-830. Pekín (China). Octubre 2000.

[5] [Macías92]
J. Macías Guarasa
Desarrollo de un sistema de reconocimiento de palabras aisladas, de gran vocabulario, dependiente del locutor, en tiempo real sobre PC
Proyecto fin de carrera. ETSIT-UPM. 1992.

[6] [Macías94a]
J. Macías-Guarasa, M.A. Leandro, J. Colás, A. Villegas, S. Aguilera y J.M. Pardo.
On the Development of a Dictation Machine for Spanish: DIVO
Proc. of the International Conference on Spoken Language Processing (ICSLP), 1994, pp. 1343-1346. 1994.

[7] [Macías94b]
J. Macías-Guarasa, M.A. Leandro, X. Menéndez Pidal, J. Colás, A. Gallardo, J.M. Pardo and S. Aguilera
Comparison of Three Approaches to Phonetic String Generation for Large Vocabulary Speech Recognition
Proc. of the International Conference on Spoken Language Processing (ICSLP), 1994, pp. 2211-2214. 1994.

[8] [Macías96]
J. Macías-Guarasa, A. Gallardo, J. Ferreiros, J.M. Pardo and L. Villarrubia
Initial Evaluation of a Preselection Module for a Flexible Large Vocabulary Speech Recognition

System in Telephone Environment
Proc. of the International Conference on Spoken Language Processing (ICSLP), 1996, pp. 1343-1346. 1996.

[9] [Macías96i]
J. Macías-Guarasa, A. Gallardo y J. Ferreiros
Módulo de búsqueda rápida para el reconocedor de habla aislada de grandes vocabularios. Manual de usuario.
Informe interno GTH-96-1. 1996

[10] [Macías99]
J. Macías-Guarasa, J. Ferreiros, A. Gallardo, R. San-Segundo, J.M. Pardo y L. Villarrubia.
Variable Preselection List Length Estimation Using Neural Networks in a Telephone Speech Hypothesis-Verification System
Proc. of the European Conference on Speech Communication and Technology (Eurospeech) 1999, Vol. 1, pp 295-298. Budapest (Hungría). Septiembre 1999

[11] [Macías00a]
J. Macías-Guarasa, J. Ferreiros, J. Colás, A. Gallardo and J.M. Pardo.
Improved Variable List Preselection List Length Estimation Using NNs in a Large Vocabulary Telephone Speech Recognition System
Proc. of the International Conference on Spoken Language Processing (ICSLP) 2000, Vol. II, pp 823-826. Pekín (China). Octubre 2000.

[12] [Macías00b]
J. Macías-Guarasa, J. Ferreiros, R. San-Segundo, J.M. Montero y J.M. Pardo
Acoustical and Lexical Based Confidence Measures for a Very Large Vocabulary Telephone Speech Hypothesis-Verification System
Proc. of the International Conference on Spoken Language Processing (ICSLP) 2000, Vol. IV, pp 446-449. Pekín (China). Octubre 2000.

[13] [Menéndez94b]
X. Menéndez Pidal, J. Macías-Guarasa, M.A. Leandro, J.A. Vallejo y J.M. Pardo
Experiments with Neural Networks in Isolated Word Recognition Tasks
Signal Processing VII: Theories and Applications. M. Holt, C. Cowan, P. Grang, W. Sandham (Eds). European Association for Signal Processing, pp. 1343-1346. Edimburgh, 1994.

[14] [SanSegundo01]
R. San-Segundo, J. Macías-Guarasa, J. Ferreiros, P. Martín, J.M. Pardo
Detection of Recognition Errors and Out of the Spelling Dictionary Names in a Spelled Name Recognizer for Spanish
Proc. of the European Conference on Speech Communication and Technology (Eurospeech) 2001. Septiembre 2001.